DL Workshop - Leveraging DL for single-cell analysis

Amos Avni, Asaf Avrahamy

September 2022

Abstract

In this work, we collaborated with prof. Ilan Tsarfaty's lab in order to leverage Deep Learning techniques for the labs' research efforts. Segmenting and tracking (S&T) single cell nucleus in high resolution videos, and time-series clustering in order to identify the different behaviors of metastatic cells with different treatments. This kind of collaboration has altered our focus from deep dive into the newest methods in the DL domain of research and sent us to explore and use tools that are utilizing DL in order to improve some of the research done in the lab on real data.
After extensive research and collaboration with Ilans' lab we were able to improve the lab's S&T pipeline in a significant order - tracking cells for much longer periods, providing new possibilities for future experiments.
As we ran an experiment from the lab through our new pipeline, we set out to discover different patterns of appearance and motion (morpho-kinetic) features using techniques from the deep-learning toolkit.
This approach, using only the data to guide us, wasn't successful in providing new insights. Nevertheless, the approach for applying deep learning on similar domains proved to be insightful, and we are hoping that further collaboration with the lab will improve our results.

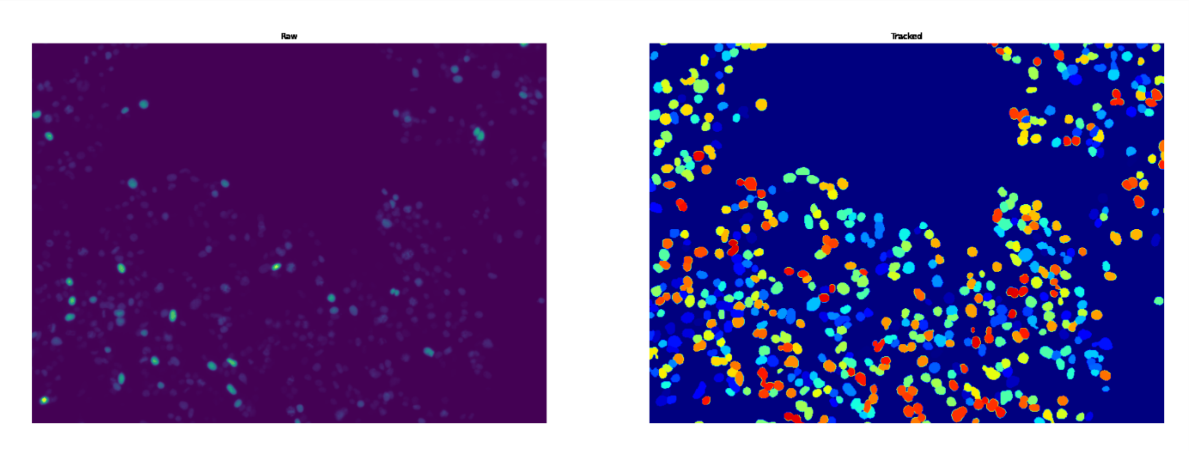
1 Introduction

1.1 Ilans' lab

The goal of the lab is to further understand the role of Met in cancer progression and metastasis, and to develop modalities for personalizing targeted Met therapy.
Metastases which are responsible for as much as 90% of cancer-associated mortality are defined as the development of secondary malignant growths at a distance from a primary site of cancer.
We focused on the analysis of the morpho-kinetic (motion and appearance) features of cells as a way to better understand the influence of the treatments on cells that are researched in the lab.

1.2 Experiments Data

Data is collected during different experiments done by the lab using a microscopic system called Incucyte. Incucyte is used for capturing live cell videos. Since the labs' experiments vary in setting and purpose, these videos vary in length, resolution, frame rate(1per_30min, 1per_hour, etc), cell density, experiment style (scatter, scratch) and staining method/gene (phase-contrast, nuclear staining, cytoplasm staining).



1.3 Stating the problem

1.3.1 Segmentation and Tracking (S&T):

The main focus of the labs' research is the morpho-kinetic behavior of the single-cell. Realizing that the current method isn't utilizing deep learning techniques, limiting its S&T abilities, we decided to start our efforts by including such techniques in the labs' pipeline, hoping to improve the labs' abilities in the field.
single cell objects in video are harder to S&T compared to other types of videos, because live cell videos have a huge time gap between frames (30 minutes to an hour compared to almost continuous in other mediums), a much denser environment (thousands of objects in each frame compared to single objects), unclear borders between objects , disappearing and reappearing objects, and little to no depth of field, object shape/color/brightness changing between frames, and objects multiplying (cell division).

1.3.2 Multivariate Time Series (MTS) Clustering:

Tracks of each cell become a time series of morphological features Major & Minor ellipse Axis , and kinetic features X & Y position (Can be transformed to displacement, acceleration and more).
The cells in each treatment are expected to have different morpho-kinetic behavior.
Labels are provided for each well and not on a single cell basis, so there is a huge variance in cell behavior in each label group. Because of that, using those labels can yield uncertain results and the lab thinks it's best to approach this problem as an unsupervised one.
The problem translates to a problem of unsupervised clustering of MTS data, and observing the different populations of clusters for each treatment.
MTS can't be clustered effectively with formal ML methods without engineered features because of temporal dimensionality, presence of complex temporal relationships, seasonality and more.

1.4 Current Method

Segmentation and tracking is done using the IMARIS program which utilizes traditional image analysis and graph algorithms. Then, morphological and kinetic features are manually extracted. Finally, PCA and hierarchical clustering is used to separate the single cells into different groups.
The manual feature extraction made by the lab led to good separation of different sub-populations of cells for each treatment. The proportion of cells in each group changes according to the genes expressed. These groups can be described using the manual morphokinetic features.

1.5 Our Addition

Segmentation and Tracking - although finding many improvements in this field we realized we wouldn't be able to train a new model by ourselves. Nevertheless, architectures with pretrained networks have shown much better results and different approaches to segmentation and tracking were examined.

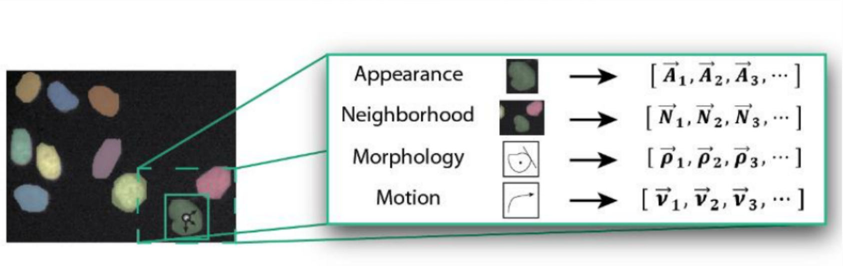
Multivariate Time Series Clustering - to improve the labs' pipeline and research tools, and discover new phenomenons hidden from traditional methods. extracting interesting patterns/behaviors from cell video data without the use of manually engineered features was our mission.
The main effort here was utilizing different DL techniques, and assessing the results in a ping-pong manner with the lab.

2 Related Work

2.1 Segmentation and Tracking of cells

[1] proposes DeepCell, a deep learning library for single-cell analysis of biological images. This library is part of a big effort led by David Valen from Caltech university, by using crowdsourcing to annotate around 1M data points. This framework is fully-supervised.
Segmentation of the nucleus of cells is done by RetinaMask models, using a ResNet50 backbone pre-trained on ImageNet. Tracking is done by a combination of deep learning and linear programming.
Object tracking is treated as a linear assignment problem. In this framework, Ni objects in frame i must be assigned to Ni+1 objects in frame i+1. In order to decide if the cell in frame i+1 is the same one from frame i, a deep learning model is characterizing each cell. This model takes information on each cell's appearance, local neighborhood, morphology, and motion. Summarizing them as a vector using a deep learning sub-model that extracts these features.
This method enables tracking cells, division of cells and disappearing and reappearing cells.

Extracting the embedded data from the sub-model seemed like a promising data compression to use later in our MTS clustering. Below is a visual demonstration of the embedding layer. Consisted of 64 variables, 16 from each branch.


Appearance
$[\vec{A}_1, \vec{A}_2, \vec{A}_3, \cdots]$
Neighborhood
$[\vec{N}_1, \vec{N}_2, \vec{N}_3, \cdots]$
Morphology
$[\vec{\rho}_1, \vec{\rho}_2, \vec{\rho}_3, \cdots]$
Motion
$[\vec{v}_1, \vec{v}_2, \vec{v}_3, \cdots]$

A supervised strategy requires a large amount of manually labeled cellular images, in which accurate segmentations and tracks at pixel level are produced by human operators, a task that is both time-consuming and requires domain experts.
New approaches, leveraging unsupervised methods from the latest image segmentation developments are achieving almost SOTA results in the segmentation field.
in [2], an alternative strategy that trains CNNs for segmentation without any human-labeled data is presented. This is achieved by first generating marker locations from nucleus images, then using a CNN to capture semantic features of whole cells.
This method allows training a CNN segmentation model without any human annotations. Only by knowing the locations of each nucleus.It can be done if the nucleus is appearing in different colors by staining, or even by traditional image analysis tools.
The image is divided into overlapping patches that are a bit larger than one cell. A UNet that minimizes the IOU between overlapping and inverse patches is being able to learn the segmentation task without supervision.
Below is the Loss function that is being minimized.
$L(M, y^1, y^2, \ldots, y^n) =- \sum_{d0,d1}\sum_i y^i_{d0,d1} - \lambda\sum_{d0,d1}\sum_{i\neq j} y^i_{d0,d1} log y^j_{d0,d1} - \beta\sum_{d0,d1}\sum_i y^i_{d0,d1}(1 - M_{d0,d1})$
Recently, the authors released an E2E framework that enables learning of segmentation with the method mentioned above[3].
Unfortunately, we haven't found completely unsupervised tracking techniques as of today.

2.2 Deep clustering of time-series data

*All of the projects below utilize an autoencoder (some with seq2seq layers) for temporal dimensionality reduction.

VRAE [4] - proposed a model that combines the strengths of RNNs and SGVB: the Variational Recurrent Auto-Encoder (VRAE). Such a model can be used for efficient, large-scale unsupervised learning on time series data, mapping the time series data to a latent vector representation. The model is generative, such that data can be generated from samples of the latent space.
VADER [5] - proposes variational deep embedding with recurrence (VaDER). VaDER relies on a Gaussian mixture variational autoencoder framework, which is further extended to model multivariate time series and directly deal with missing values.
DCEC [6] - proposes a deep convolutional embedded clustering algorithm. using a convolutional autoencoder structure in which a clustering oriented loss is directly built on the learned embedded features, the reconstruction loss of convolutional autoencoders and the clustering loss are simultaneously minimized to jointly perform feature refinement and cluster assignment.
DTC [7] - proposes a novel temporal clustering layer for cluster assignment. Then it jointly optimizes the clustering objective and the dimensionality reduction objective.
DTCR [8] - integrates the temporal reconstruction and K-means objective into the seq2seq model.
[9] proposes a supervised auto-encoder that jointly predicts classes using labels and reconstructs inputs. using classification as a task of a second network head during the entire training process and not an auxiliary loss.

2.3 Deep clustering and classification of trajectories

As the data we've extracted is composed of trajectory and morphology data, we decided to split the problem into smaller segments, focusing on clustering the trajectory data of each cell.
Trajectory clustering is discussed over several domains, GPS trajectories of cars, aerocraft trajectories, human trajectories, and more.

In [10] for the purpose of classifying treated/untreated cancer cells, after tracking single cells, a plot of the trajectory was painted visually, then fed into a retrained AlexNET ConvNet through transfer learning on the cells data. This approach, although giving good results for classification, is neglecting the time dimension and morphology dimension. This gives us a good signal that feeding neural nets with trajectory data, may help to better understand different behaviors of cells.

In [11] Feeding airplane flow trajectories into CNN autoencoders to extract information from the latent space while optimizing both the reconstruction error and a clustering loss term gave good clustering results. In this paper the authors utilize DCEC in order to cluster trajectories. We have noticed that the clustering was highly affected from spatio-temporal features, start and end point, features that aren't relevant in our data.

In [12] there was added a preprocessing phase of the data to capture space and time **invariant** characteristics of the trajectories. This approach allows this paper to cluster trajectories by the character of the motion (straight, bending, circular), and less by their start and end position, direction of movement. This is important especially in our case , where particles are located in different parts of the image and move in various directions - as the absolute location nor the absolute direction of movement is important. This method utilized a seq2seq autoencoder after the preprocessing phase that led to good results on simulated and real data.

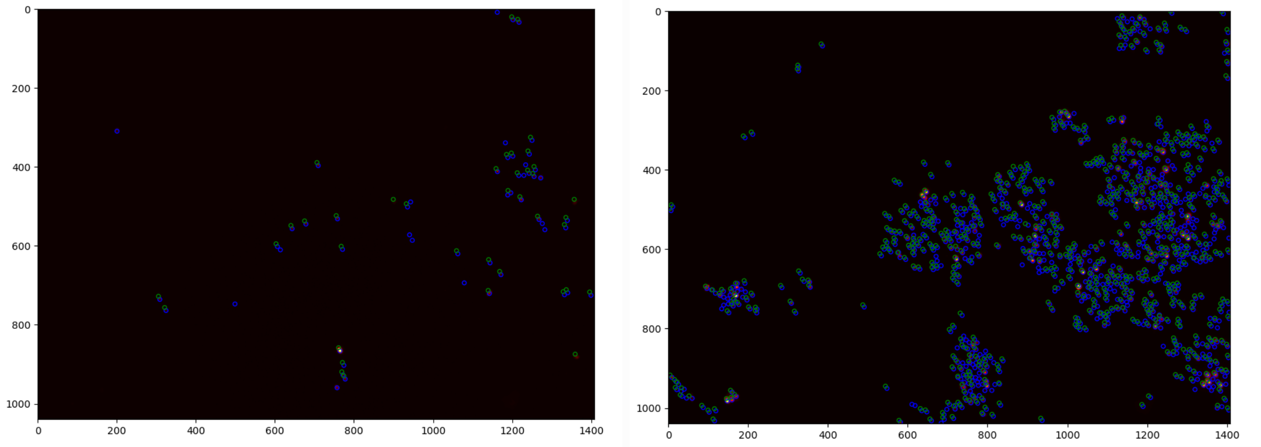
3 Experiment

We used data from an experiment named JH01, conducted in June 2022.
In the experiment 5 different treatments were applied to different groups of cells - called wells.
Each treatment was applied to 3 different wells. Each well contains ~5000 cells.
Each well is separated by the incucyte into 4 views with a resolution of 1480*1040.
Each well was captured every 30 minutes, for 95 frames.

3.1 TIF to Tracking

We've segmented and tracked the cells in the wells using DeepCell framework. We used a GPU as it is an exhaustive task - around 1000 cells in each view, and around 100 frames per view, 4 views per well, around 30 wells in total. (Segmenting around 12M cells)
We have segmented almost every cell that was segmented by Imaris, and most of the time found more cells in each frame.
We've been able to track the cells for much longer periods.
An average length of 30 frames compared to average length of 6 from Imaris.

We modified the framework to extract more features during the inference stage of the network -
cell centroids, morphological features and the embedding layer (neighborhood encoder, cell appearance, kinetic features and morphology).



images comparing our cell detection (blue) vs imaris (green) right: frame15 left: frame90 notice how the number of cells increases dramatically, this is because the fluorescence staining method becomes more active over time. And also because cells are multiplying.

3.2 Processing Tracking Data

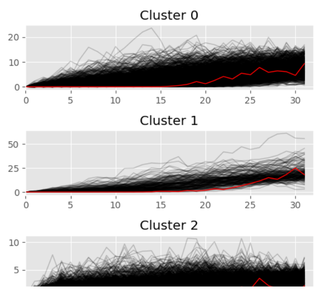
At this point, image data becomes multivariate time series data. the data is a grid of size total_observed_cells*frames (~$3000 * 95 * 69$) for each view of a well.
Each observed cell has features that were extracted from the TIF files - [X,Y,Area,Minor axis,Major-Axis,Embedding]. These features are not null in all frames where the cell was tracked, and are all null in frames where the cell is missing.
First, we chose to split each cells' row into continuous track objects.
Since continuous tracks of different length exist, we chose to select a minimum length (16,32) of tracks we are interested in, then we dropped all shorter tracks. next we cut all longer tracks to the desired length and end up with a grid of $(size\ cells\ *\ minimum\ length\ *\ 69)$
For each of those continuous track objects, we paired the well name ('D2','G4',etc.) and the treatment type (Gab1,Grb2,control,etc.)
Then we shifted all trajectories to start at (0,0), and normalized each feature using a standard scaler.
*multiple different tracks of the same cell can exist with this method of preprocessing, since we are not using the cells' id and only use well-level labels this only adds more data to work with.

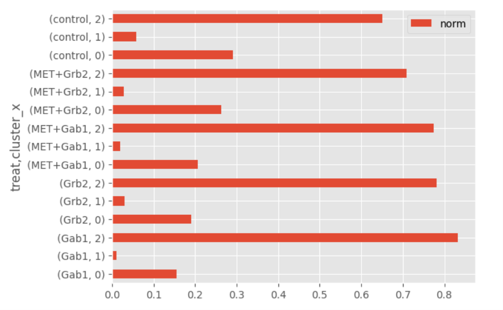
3.3 Tracking Data to Clustering

Instead of extracting features manually, we would like to feed each continuous track vector (MTS) into a neural network that will try to cluster each track by its morpho-kinetic signature.
In order to accomplish this we used an autoencoder as a way to 'extract' hidden features of trajectory and morphological features, and used deep clustering techniques in order to find the most informational clusters.

3.3.1 Baseline -Univariate Time Series clustering

In the lab today, manual features of the morpho-kinetic behavior of cells are used to differentiate between the different treatments.
We have tried to use univariate time series clustering to check if we can also cluster by these features.
For example, displacement feature- how far the cell has traveled from its original point.
Here using dynamic time warping distance metric we divided the data into 3 clusters.
This metric is much more meaningful than the euclidean distance metric (explanation below).
Upper figure - clusters found by time-series k-means.
The Y axis represents the value of displacement of each cell by time.
Cluster 3 represents the group of cells that are staying relatively closer to their starting point, and cluster 2 represents cells that are traveling far. In the lab this behavior has biological meaning.
Bottom figure - distribution of clusters in each treatment. This is the method in which we can try to explain the different behaviors of each treatment.
This method can't be expanded to multivariate features and therefore is just for the baseline.





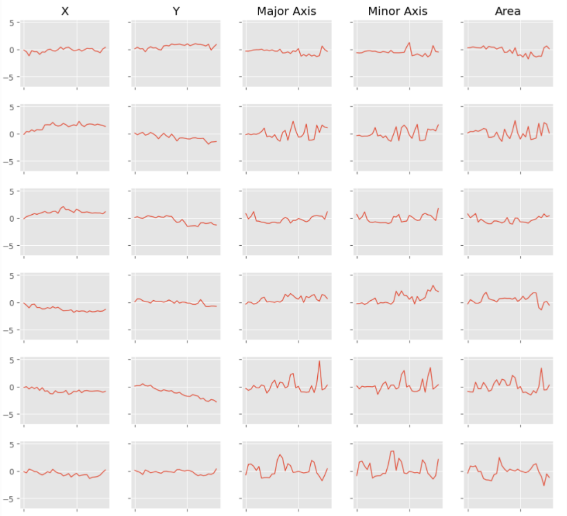
Dynamic time warping is a method that originates from the speech recognition field. It aims at aligning two sequences of feature vectors by solving a dynamic programming problem of the warp function. It allows to find similar patterns between temporal behavior which are not captured by euclidean distance. It is quadratic in time without optimizations.

3.3.2 Autoencoders - unsupervised clustering

Although labels of treatment of each well are available, we were guided to discard them and start our efforts with clustering the morpho-kinetic behavior of the cells without supervision.
This is backed up by domain knowledge that the treatments aren't changing all cells behavior and therefore the labels are noisy and can be misleading.
In order to cluster the multivariate time series data, we decided to use autoencoders as a way to extract latent features. Noticing most of the literature around this subject used such architectures. Hoping they will reveal useful patterns of movements in different cell groups.

Our goal was to optimize reconstruction loss while maintaining a low dimension bottleneck. Using the bottleneck to extract good latent representation of the trajectories on which we can cluster the multivariate time series.

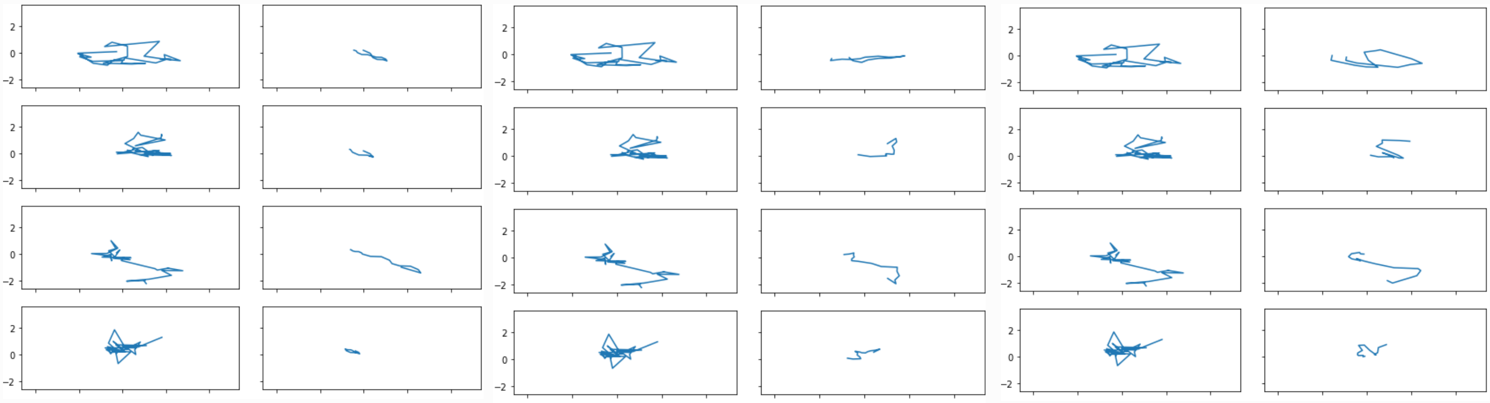
Using the morphological features - Major axis, Minor axis, Area, hindered the performance of every model we tried.
Noticing how volatile these features are, we decided to exclude them from our efforts.
The reason for the huge “choppiness” of these features is unknown to us, and may be a product of the laser imagery, the staining technique used or other mechanical factors, since this behavior is also present in the labs’ Imaris data.



After experimenting with different architectures, we chose to implement an asymmetric auto-encoder as proposed in [DTC 8].
Using stacked conv1D layers as a way to capture complex patterns of the features, followed by two bidirectional LSTMs as the encoder, and upsampling followed by a Transposed conv2D as the decoder.

Evaluating both CNN and RNN as encoding units to extract time based features we empirically saw better results using the seq2seq with LSTM instead of stacked conv2d layers.

This architecture allowed us to reach the minimal reconstruction loss with low dimension bottleneck, and effectively reduce the ‘time’ dimension from our data.
Left to right figures - varying bottlenecks and samples of the encoder prediction of 32 length tracks with X,Y positions - 4,8,16 respectively.



After training, we reduced our data to the latent dimension and clustered it via PCA,TSNE.
In addition, we plotted representatives from each cluster, our results were disappointing. We didn't observe any noticable difference between the different clusters, coloring them by our cell-type labels resulted in a uniform scatter.

As our mission was to cluster the latent layer, after pretraining the autoencoder, we used the architecture defined in DTC , that uses 2 heads, one for autoencoding and one for clustering, in order to better separate the latent layer for clustering.
The network optimizes the reconstruction loss with MSE loss, and the clustering loss with KL-Divergence loss.

$$q_{ij} = \frac{\left(1 + \frac{siml(z_i,w_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}}{\sum_{j=1}^{k}\left(1 + \frac{siml(z_i,w_j)}{\alpha}\right)^{-\frac{\alpha+1}{2}}}$$

Clustering loss is defined as minimizing the KL-Divergence loss between q and p.

$q_{i,j}$- the probability of input i to belong to cluster j. q is calculated according to different similarity metrics that are suggested in the paper, these similarities start with euclidean distance and continue with correlation and auto-correlation.

$p_{i,j}$- is a squared sum combination of $q_{i,j}$.

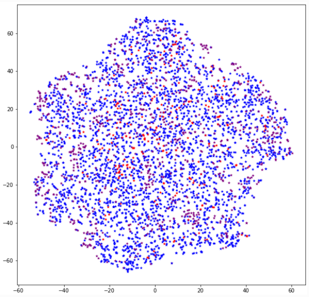
$$p_{ij} = \frac{q_{ij}^2 / f_j}{\sum_{j=1}^{k} q_{ij}^2 / f_j}$$

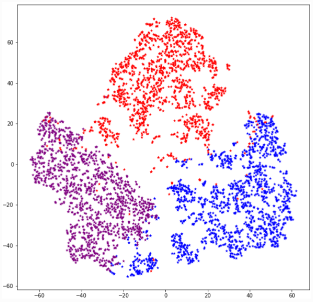
This approach increased our overall reconstruction loss while significantly improving the clustering loss - therefore maximizing separation of clusters in the latent dimension.

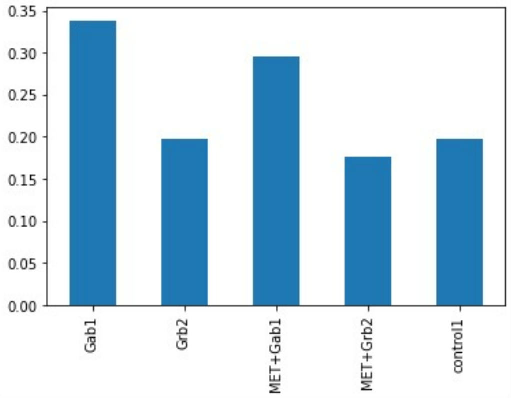
Left figure - TSNE representation of the latent layer predicted by the autoencoder before training with clustering loss.

Middle figure - TSNE representation after training with clustering loss.

Right figure - We were able to see different representations of treatments in a specific cluster. Although not significant enough , this kind of approach can open further investigation of the patterns of behavior in the specific cluster that is 'marking' different treatments.







3.3.3 Supervised classification

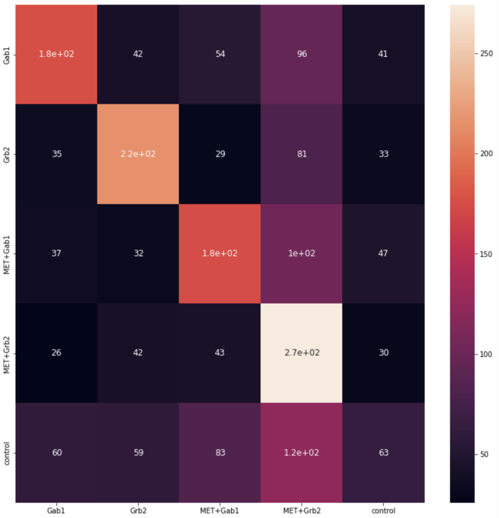
After our unsuccessful venture into unsupervised training methods, and going over the results with the lab, we decided to try a supervised approach, trying to classify based on well-treatment labels .

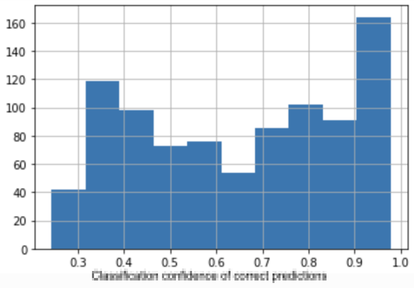
- each group contains a large proportion of control cells and a big variance in cell behavior is present (so labels do not dictate clear behavior)

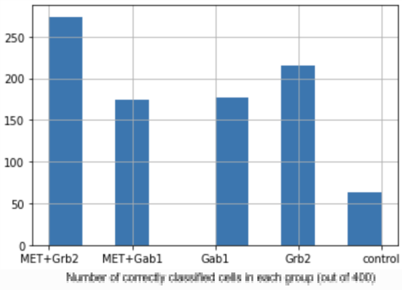
using the same encoder architecture, now with a single classification objective. using the accuracy and auc and optimizing the model performance with k fold cv, we achieve an accuracy of 0.43-0.46 and an auc of ~0.75 for five classes.

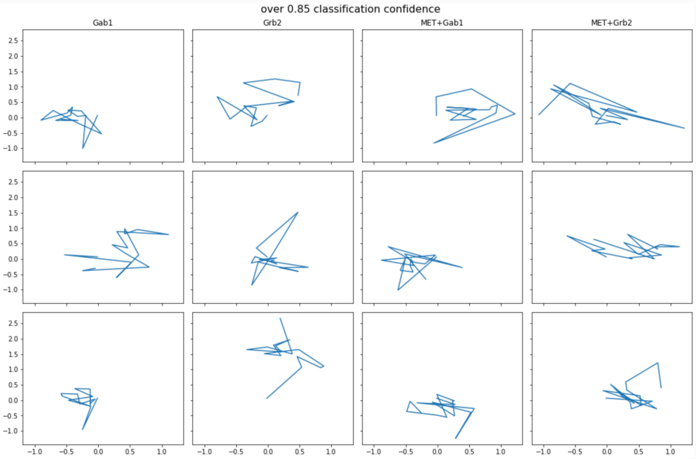
We then plot the confusion matrix and representatives from each class who the network predicted correctly with high confidence, and those it has predicted correctly with low confidence (by chance), working with the lab to assess the difference between classes as seen by the network.

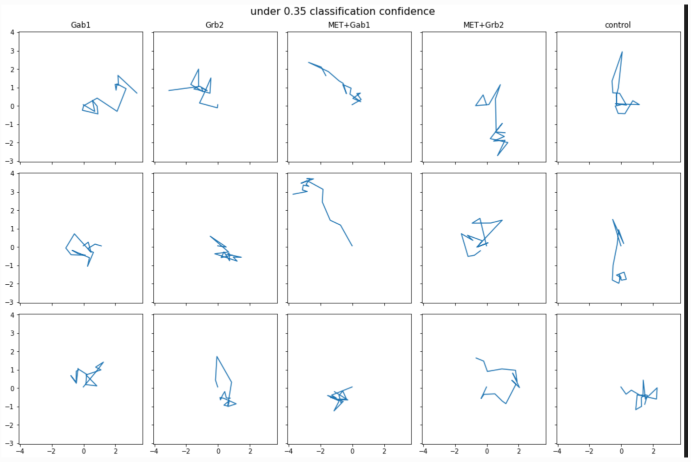
We noticed the network confuses the cells under 'control' uniformly with the other groups, this is backed up by the information that each of the other groups contain a large population of control cells (cells that do not respond to the treatment). Contrary to that, each of the treatment groups contain a subpopulation on which the model predicts correctly with high confidence.











5 Future Work

5.1 Future work

Applying unsupervised segmentation of cytoplasm staining data in other experiments in Ilan’s lab using methods we’ve found [2][3]. These methods proved to provide better segmentation on the data we used (nuclear staining) compared to Deepcell and can be used for more difficult data (that we have un-annotated and can’t train on). Implementing this method in our pipeline as the segmentation part while keeping Deepcell as the tracking infrastructure will improve overall results and open the way for more experiments in the lab.

Adding more live cell video data to our dataset for further improving our models.

Using DL trajectory analysis methods to extract more complex features for clustering from our X,Y data which proved to be useful in classifying cell types.

Since cells can change behavior/class over time, building a model with the task of spotting change of behavior (models of movements like Levy flight and Brownian motion)

GitHub repository:
https://github.com/amosaNN/workshop-in-deeplearning

References

1. Erick Moen, David Van Valen. Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning https://doi.org/10.1101/803205
2. Din, N.U., Yu, J. Training a deep learning model for single-cell segmentation without manual annotation. Sci Rep 11, 23995 (2021). https://doi.org/10.1038/s41598-021-03299-4
3. Prem Shrestha, Nicholas Kuang, Ji Yu. Efficient End-to-end Learning for Cell Segmentation with Machine Generated Incomplete Annotations https://doi.org/10.1101/2022.07.03.498609
4. Otto Fabius & Joost R. van Amersfoort, VARIATIONAL RECURRENT AUTO-ENCODERS https://arxiv.org/pdf/1412.6581.pdf https://github.com/tejaslodaya/timeseries-clustering-vae
5. Johann de Jong, Mohammad Asif Emon, Ping Wu, Reagon Karki, Meemansa Sood, Patrice Godard, Ashar Ahmad, Henri Vrooman, Martin Hofmann-Apitius, Holger Fröhlich, Deep learning for clustering of multivariate clinical patient trajectories with missing values, *GigaScience*, Volume 8, Issue 11, November 2019, giz134, https://doi.org/10.1093/gigascience/giz134
6. Xifeng Guo, Xinwang Liu, En Zhu, Jianping Yin. Deep Clustering with Convolutional Autoencoders. ICONIP 2017. https://xifengguo.github.io/papers/ICONIP17-DCEC.pdf
7. Madiraju, N. S., Sadat, S. M., Fisher, D., & Karimabadi, H. (2018). Deep Temporal Clustering : Fully Unsupervised Learning of Time-Domain Features. http://arxiv.org/abs/1802.01059
8. Qianli Ma, Jiawei Zheng, Sen Li, Gary W. Cottrell, Learning Representations for Time Series Clustering, (NeurIPS 2019). https://proceedings.neurips.cc/paper/2019/file/1359aa933b48b754a2f54adb688bfa77-Paper.pdf.
9. Lei Le, Andrew Patterson, Martha White, “Supervised autoencoders: Improving generalization performance with unsupervised regularizers”, (NeurIPS 2018), https://papers.nips.cc/paper/2018/file/2a38a4a9316c49e5a833517c45d31070-Paper.pdf
10. Mencattini, A., Di Giuseppe, D., Comes, M.C. et al. Discovering the hidden messages within cell trajectories using a deep learning approach for in vitro evaluation of cancer drug treatments. Sci Rep 10, 7653 (2020). https://doi.org/10.1038/s41598-020-64246-3
11. Xavier Olive, Luis Basora, Benoit Viry, Richard Alligier. Deep Trajectory Clustering with Autoencoders. ICRAT 2020, 9th International Conference for Research in Air Transportation, Sep 2020, Tampa, United States. ffhal-02916241f https://hal-enac.archives-ouvertes.fr/hal-02916241
12. D. Yao, C. Zhang, Z. Zhu, J. Huang and J. Bi, "Trajectory clustering via deep representation learning," 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 3880-3887, https://doi.org/10.1109/IJCNN.2017.7966345
13. Pierre Baldi. 2011. Autoencoders, unsupervised learning and deep architectures. In Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop - Volume 27 (UTLW'11). JMLR.org, 37–50.
14. Lafabregue, B., Weber, J., Gançarski, P. *et al.* End-to-end deep representation learning for time series clustering: a comparative study. *Data Min Knowl Disc* 36, 29–81 (2022). https://doi.org/10.1007/s10618-021-00796-y
15. Ilan Tsarfaty, Or Megides, Ohad Doron, Hagar Alaloof, Ori Moskowitz, Judith Horev; Abstract PO-089: Morphokinetic single-cell analysis and machine learning as a tool to characterize breast cancer cell motility and response to therapy. *Clin Cancer Res* 1 March 2021; 27 (5_Supplement): PO–089. https://doi.org/10.1158/1557-3265.ADI21-PO-089
16. Ali Alqahtan, Mohammed Ali, Xianghua Xie and Mark W. Jones, Deep Time-Series Clustering: A Review https://www.mdpi.com/2079-9292/10/23/3001/pdf
17. Xifeng Guo, Xinwang Liu, En Zhu, Jianping Yin. Deep Clustering with Convolutional Autoencoders. ICONIP 2017. https://xifengguo.github.io/papers/ICONIP17-DCEC.pdf
18. Kazuya Nishimura, Hyeonwoo Cho, Ryoma Bise, Semi-supervised Cell Detection in Time-lapse Images Using Temporal Consistency, arXiv:2107.08639 [cs.CV] 19 Jul 2021 https://arxiv.org/pdf/2107.08639.pdf
19. Kazuya Nishimura, Junya Hayashida, Chenyang Wang, Dai Fei Elmer Ker, Ryoma Bise, Weakly-Supervised Cell Tracking via Backward-and-Forward Propagation, arXiv:2007.15258v1 [cs.CV] 30 Jul 2020 https://arxiv.org/pdf/2007.15258.pdf
20. J. Hayashida, K. Nishimura and R. Bise, "Consistent Cell Tracking in Multi-frames with Spatio-Temporal Context by Object-Level Warping Loss," 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 1759-1768, doi: 10.1109/WACV51458.2022.00182. https://openaccess.thecvf.com/content/WACV2022/papers/Hayashida_Consistent_Cell_Tracking_in_Multi-Frames_With_Spatio-Temporal_Context_by_Object-Level_WACV_2022_paper.pdf
21. Yang L, Ghosh RP, Franklin JM, Chen S, You C, Narayan RR, et al. (2020) NuSeT: A deep learning tool for reliably separating and analyzing crowded cells. PLoS Comput Biol 16(9): e1008193. https://doi.org/10.1371/journal.pcbi.1008193
22. Junya Hayashida, Kazuya Nishimura, Ryoma Bise, “MPM: Joint Representation of Motion and Position Map for Cell Tracking”, arXiv:2002.10749v2 [cs.CV] 26 Feb 2020 https://arxiv.org/pdf/2002.10749v2.pdf

Appendix

1 Biology Domain Knowledge

During metastatic dissemination, a cancer cell from a primary tumor invades the surrounding tissue, enters the microvasculature of the lymph and blood systems, survives and translocates mainly through the bloodstream to microvessels of distant tissues, exits from the bloodstream, survives in the microenvironment of distant tissues and finally adapts to the foreign microenvironment of these tissues in ways that facilitate cell proliferation and the formation of a macroscopic secondary tumor.

Cancer cells were shown to metastasize using amoeboid cell motility mode. Amoeboid cell motility is a crawling-like movement accomplished by protrusion of the cytoplasm of the cell involving the formation of pseudopodia. The cytoplasm slides and forms a pseudopodium in front to move the cell forward.

Met transduces signaling mainly through the adaptor proteins Gab1 and Grb2. To further study the molecular mechanisms involved in Met-induced amoeboid motility, we examined the role of these adaptors in this phenomenon. HEK293T expression of WT-Met alone or with Gab1 or Grb2 demonstrated that Met signaling through Gab1 increased immediate amoeboid-blebbing motility while Met signaling through Grb2 induced delayed amoeboid filopodia motility compared to Met. Examination of the dynamics of Gab1 and Grb2 phosphorylations revealed that Met-induced Gab1 phosphorylation was immediately and paralleled with Met's phosphorylation, while Met-induced Grb2 phosphorylation was delayed. This delay was compatible with the delayed appearance of the amoeboid filopodia phenotype and indicated that this event is probably transcriptionally regulated.

Study of the Met-RTKs co-activation by examination of the RTKs phosphorylation levels followed by the study of the signaling pathways by assessment of the phosphorylation levels of cytoplasmic kinases and analysis of transcriptional activation of cell signaling pathways targets revealed that Met+Gab1 significantly increased RTKs co-activation, while Met+Grb2 significantly decreased RTKs co-activation as compare to Met. These changes in amoeboid-blebbing motility phenotype presented in Met+Gab1 compared to Met

In conclusion, this study contributed to the knowledge that Met's molecular mechanism induces amoeboid cell motility. This mechanism includes RTKs co-activation, which probably characterizes induction of amoeboid cell motility and will help to develop future strategies that will inhibit cell motility leading to generating therapies that target metastasis. This understanding can also contribute to the changing of personalized, targeted therapy, based on the inhibition of one receptor, to simultaneous inhibition of several co-activating receptors.