Machine Learning Pipelines for Deconvolution of Cellular and Subcellular Heterogeneity from Cell Imaging

A dissertation

Submitted to the faculty of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirement for the Degree of

Doctor of Philosophy in

Biomedical Engineering

August 6th, 2019

by

Chuangqi Wang

APPROVED:

Kwonmoo Lee, Ph. D.
Assistant Professor, Advisor
Department of Biomedical Engineering
Worcester Polytechnic Institute

Songbai Ji, Ph. D.
Associate Professor, Committee Chair
Department of Biomedical Engineering
Worcester Polytechnic Institute

Dirk Albrecht, Ph. D.
Associate Professor
Department of Biomedical Engineering
Worcester Polytechnic Institute

Dmitry Korkin, Ph. D.
Associate Professor
Department of Computer Science
Director of Bioinformatic and Computational Biology
Worcester Polytechnic Institute

Hakho Lee, Ph. D.
Associate Professor in Radiology, Harvard Medical School
Director of the Biomedical Engineering Program at the Center for Systems Biology
Massachusetts General Hospital (MGH)

Abstract

Cell-to-cell variations and intracellular processes such as cytoskeletal organization and organelle dynamics exhibit massive heterogeneity. Advances in imaging and optics have enabled researchers to access spatiotemporal information in living cells efficiently. Even though current imaging technologies allow us to acquire an unprecedented amount of cell images, it is challenging to extract valuable information from the massive and complex dataset to interpret heterogeneous biological processes. Machine learning (ML), referring to a set of computational tools to acquire knowledge from data, provides promising solutions to meet this challenge. In this dissertation, we developed ML pipelines for deconvolution of subcellular protrusion heterogeneity from live cell imaging and molecular diagnostic from lens-free digital in-line holography (LDIH) imaging.

Cell protrusion is driven by spatiotemporally fluctuating actin assembly processes and is morphodynamically heterogeneous at the subcellular level. Elucidating the underlying molecular dynamics associated with subcellular protrusion heterogeneity is crucial to understanding the biology of cellular movement. Traditional ensemble averaging methods without characterizing the heterogeneity could mask important activities. Therefore, we established an ACF (auto-correlation function) based time series clustering pipeline called HACKS (deconvolution of heterogeneous activities in coordination of cytoskeleton at the subcellular level) to identify distinct subcellular lamellipodial protrusion phenotypes with their underlying actin regulator dynamics from live cell imaging. Using our method, we discover “accelerating protrusion”, which is driven by the temporally ordered coordination of Arp2/3 and VASP activities. Furthermore, deriving the merits of ML, especially Deep Learning (DL) to learn features automatically, we advanced our pipeline to learn fine-grained temporal features by integrating the prior ML analysis results with bi-LSTM (bi-direction long-short term memory) autoencoders to dissect variable-length time series protrusion heterogeneity. By applying it to subcellular protrusion dynamics in pharmacologically and metabolically perturbed epithelial cells, we discovered fine differential response of protrusion dynamics specific to each perturbation. This provides an analytical framework for detailed and quantitative understanding of molecular mechanisms hidden in their heterogeneity.

Lens-free digital in-line holography (LDIH) is a promising microscopic tool that overcomes several drawbacks (e.g., limited field of view) of traditional lens-based microscopy. Numerical reconstruction for hologram images from large-field-of-view LDIH is extremely time-consuming. Until now, there are no effective manual-design features to interpret the lateral and depth information from complex diffraction patterns in hologram images directly, which limits LDIH utility for point-of-care applications. Inherited from advantages of DL to learn generalized features automatically, we proposed a deep transfer learning (DTL)-based approach to process LDIH images without reconstruction in the context of cellular analysis. Specifically, using the raw holograms as input, the features extracted from a well-

trained network were able to classify cell categories according to the number of cell-bounded microbeads, which performance was comparable with that of object images as input. Combined with the developed DTL approach, LDIH could be realized as a low-cost, portable tool for point-of-care diagnostics.

In summary, this dissertation demonstrate that ML applied to cell imaging can successfully dissect subcellular heterogeneity and perform cell-based diagnosis. We expect that our study will be able to make significant contributions to data-driven cell biological research.

To my parents, elder sister who loved me unconditionally,

My wife, Dr. Fan Zhang, who always fully supported me,

And my gorgeous son, Ian, who spread joy and positive change in my life.

Acknowledgements

The life to presume a Ph.D. is an exciting adventure with laugh and tears. It will be more challenging without full-supporters. Here, it's my pleasure to acknowledge many people, who helped me in my graduate study and research at WPI.

First, I would like to express my highest gratitude to my advisor, Dr. Kwonmoo Lee, who always is supportive and ready to discuss my research projects when I requested over the past four years. He introduced me into the field of microscopy imaging and cell migration hand by hand. He taught me how to distinguish different types of cell images using the characteristics of different microscopes and how to interpret the cell motility data sets and associate the motility profile with regulator dynamics by visually and correlation analysis. Also, he guided me to explore all the research steps including how to generate reasonable hypotheses, how to generate data to validate the hypotheses, how to make an attractive story to write papers and present my work at conferences. I really enjoy working with him of discussing all potential research directions. His motivation and dedication in developing machine learning strategies to reveal underlying mechanisms in cell motility and drug discovery has been a constant inspiration for my progression in scientific research.

I appreciate my committee members for their valuable comments and time commitment. Thank Dr. Dirk Albrecht, as my qualifying exam and dissertation committee member for his critical comments on my dissertation proposal. From his biomedical imaging analysis course, I had a wonderful experience to build up a microscopy from scratch, which helped me understand the mechanism of microscopy. I thank Dr. Songbai Ji for his support and comments on method justifications for my dissertation proposal. I'd like to thank Dr. Dmitry Korkin, who agreed to be my dissertation committee member. His valuable comments on my machine learning algorithms were very helpful. Moreover, I would like to recognize Dr. Hakho Lee, who co-mentored me on the molecular diagnostic project using lens-free digital in-line holography (LDIH) technique. We have an amazing collaboration and published a paper based on this project.

I would also like to thank Dr. Patrick Flaherty, who first introduced me to the field of biomedical engineering and has offered me an opportunity in his lab as research associate. I really appreciate that he set up a micro-bio project using deep learning techniques five years ago and led me into this cutting-edge research filed. Also, I thank him for the strong recommendation during my PhD application.

Beyond this, I learned a lot from our lab members and other graduate students in the biomedical engineering department. First, I'd like to show my gratitude to Dr. Heejune Choi, my close collaborator. I am quite impressive about her profound biology knowledge and critical thinking. Without her contribution, my projects could not progress smooth and fast. I also learned a lot from her about writing skill and figure organization and visualization. I also thank Xitong Zhang, Yundong Yu, Joseph Tadros, Bing Zhao, Xiang

Pan, Sung-Jin Kim, Yenyu Chen for positive supports and valuable comments. I also want to thank Dan Lawler because he helped me polish my writing in my proposal. Moreover, I'd like to show my thanks for my REU mentee Lucy Woodbury and Tessa Curtis, two smart undergraduate students, who contributed to test some hypothesis of my projects.

At last and most importantly, I'd like to acknowledge my wife, Dr. Fan Zhang, who also graduated from BME at WPI. Without her support and positive encouragement, I would not be able to overcome all the challenges throughout my PhD study. Special thanks are given to her for her willingness and supervision when things became messy. Also, I'd like to thank my son, Ian, who brought a lot of joy and smile during the past four years. Furthermore, I must express my gratitude to my parents and elder sister for their endless love and support. Without their consistent encouragement, I couldn't keep making progress in the future.

Content

Figures

Tables

Chapter 1

Introduction

In past decades, high-throughput imaging techniques have provided new big data opportunities for biological research[1-4]. With the advent of these advanced techniques, vast amount of different types of large-scale data [5-6]could be generated in a cost-efficient manner. However, this poised a new challenge with handling, processing and interpreting data using traditional analysis pipeline. Furthermore, biological systems are always far more complex and heterogeneous than we can expect[7]. Technological advances also provide the possibility to study more complex biological phenomena. Particularly, imaging technologies[8] have allowed the researchers to capture the changes at varying biological scales, such as collective cell activities, single cell phenomenon and even coordination of different regulators at the subcellular scale in a high-throughput manner. However, it is difficult to analyze these complex biological phenomena due to the massive variation from heterogeneous biological system and extra noise from the data acquirement process. Therefore, there is an unmet need for more advanced analysis pipeline to handle with those massive and complex datasets. Machine learning (ML)[9-10], referring to a field of study that gives computers the ability to learn without being explicitly programmed, quoted from Arthur Samuel, a pioneer in ML, provides promising solution to meet this challenge[22-23]. ML aims to provide a generalized and automatic pipeline to mine the complex dataset to discover new knowledge and achieves a great progress in different applications[11-13] in past decades like imaging process[11], robotics[13] and biomedical data analysis[12]. The well-known example should be AlphaGo[13], a ML-based computer program to beat many famous human professional Go players. Moreover, in biological field, Prof. Baker utilized ML techniques to predict and design the three-dimensional structures of proteins[12]. However, most of ML techniques highly rely on the manually designed feature from feature engineering using the domain knowledge from experts. Quoted from Andrew Ng, a Professor in ML from Stanford University, “Feature engineering is difficult, time-consuming, and requires expert knowledge”[14]. It substantially hinders the application of ML in biological and biomedical field because of challenges to transfer biological knowledge to engineer. In past 10 years, Deep Learning (DL)[15-16], as a sub-field of ML, mimicking neural networks to learn effective representation with multiple level of abstraction hierarchically, become widely accepted and revolutionized many fields[17-19], especially in imaging process and interpretation. DL techniques, foregoing the need for rate-limiting feature engineering, have great potential to contribute to understanding of the mechanism of biological systems[20-21].

In this dissertation, we explored different ML techniques including DL to interpret various cell imaging dataset, which could be produced at the throughput of hundreds of images per day. We successfully demonstrated the power of ML in interpreting the cell imaging dataset and deconvolving heterogeneity involved in the biological system. First, to extract data representation automatically using DL in the context of cellular analysis, we proposed a deep transfer learning (DTL)-based approach to extract

features from hologram images generated by large-field-of-view lens-free in-line holography (LDIH) for cell categories classification according to the number of cell-bounded microbeads. We demonstrated that the performance using hologram image is comparable to that from object image. Second, I developed an ACF (auto-correlation function)-based machine learning pipeline to identify distinct subcellular lamellipodial protrusion phenotypes with their underlying actin regulator dynamics from live cell imaging. Cell protrusion is involved in many different biological and pathological processes such as wound healing, immune response, embryonic development and even cancer metastasis. Third, by leveraging the capability of DL to learn important features automatically from dataset, I advance our pipeline to learn fine-grained temporal feature directly from time series protrusion velocity time series based on Guided bi-LSTM (bi-direction long-short term memory) autoencoder. Using this pipeline, we successfully demonstrate that the integration of conventional ML and DL enables us to quantitatively understand the biological mechanisms hidden in their heterogeneity. In section 1.1, I will explain the fundamental biological questions we focused on in this thesis in detail. In section 1.2, I will introduce the basic ML background and reviews the current ML techniques involved in time series analysis.

1.1 Problem Statement

In this dissertation, we mainly focus on two problems: molecular diagnostics based on the classification of diffraction pattern from in-line holography and deconvolution of subcellular protrusion heterogeneity from live cell imaging.

1.1.1 Molecular diagnostics based on the classification of diffraction pattern from in-line holography

Lens-free digital in-line holography (LDIH)[24-25] has become a new tool for biological application. It has several advantages over conventional lens-based light microscopy such as simple hardware construction and large field of view. However, complex diffraction patterns directly recorded from LDIH could not be recognized by human. Therefore, reconstruction is always required, but this requires costly computation sources, hindering wide application of LDIH particularly in resource limited settings. Therefore, how to effectively reduce this computation burden become an important issue. In this dissertation, we hypothesis that diffraction patterns could be well recognized by machine automatically even though human cannot interpret visually if effective features are used. We explore the possibility of extracting effective feature information directly from diffraction patterns recorded in hologram image which forgoes the time-consuming reconstruction step. Using the feature representation from well-trained deep learning model in image classification, we successfully built a diffraction pattern-based classifier which predict the number of the microbeads attached to cancer cells.

1.1.2 Deconvolution of subcellular protrusion heterogeneity from live cell imaging

Cell migration is involved in many different biological and pathological processes such as wound healing, immune response, embryonic development and even cancer metastasis. Cell protrusion, which initiates cell migration, determines the directionality and persistence of cell movements and facilitates the exploration of the surrounding environment[26-28]. Many researches have been investigated cell protrusion and found that heterogeneous motility at the cellular level existed in different cell types and vitro environments[29-31]. It is observed that at the subcellular level cell protrusion is morpho-dynamically heterogeneous (Fig. 1.1) [32-33] and occurs over varying periods of time, thus creating a heterogeneous temporal length. At different spatial location of cell boundary, as shown in Fig 1.1, the leading-edge dynamics are highly variable and ever changing in the same location. We call this phenomenon "Subcellular Protrusion Heterogeneity"[34]. Elucidating the subcellular protrusion heterogeneity are crucial to understanding cellular movement and even more complex phenomena such as collective cell migration.

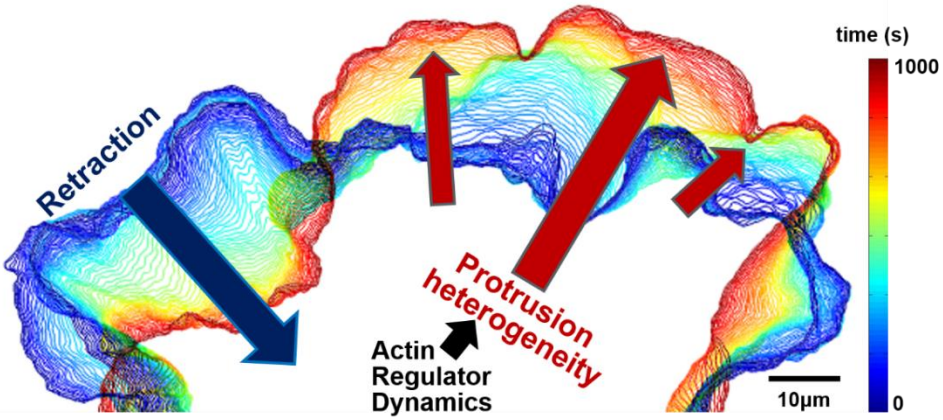


Fig. 1.1 Heterogeneity of cell protrusion. (In courtesy of Hee June Choi).

Furthermore, cell protrusion is driven by actin polymerization that is nonlinearly coordinated by multiple regulators at micron and sub-minute scales[35-38]. Hundreds of actin regulators are coordinated to organize actin cytoskeleton. For example, Arp2/3, a major actin nucleator generating new branch filaments from existed actin filament, is widely accepted to play an important role in assembling actin networks to push the membrane forward. Moreover, many other actin regulators such as formins (mDia)[39] also play roles in actin nucleation. In addition, actin elongators such as vasodilator stimulated phosphoprotein (VASP)[40] prevents binding of capping proteins to barbed ends of actin filaments and recruit more actin monomers to the tip of actin filament through profilin, which is known to be important in cell protrusion. Furthermore, actin depolymerizer such as cofilin severs existing actin filaments, and capping proteins terminate actin elongation.

Even though the role of each actin regulator is well known, it is still unclear that how these regulators are spatiotemporally coordinated to drive the heterogeneity of protrusion[41-42]. To understand this complex system, we proposed our central hypothesis that subcellular protrusion is spatiotemporally regulated by

distinct molecular dynamics and differentially affected by pharmacological perturbations. To evaluate this hypothesis, we asked three questions to be answered as follows:

1): What are the hidden protrusion patterns at the subcellular level?

2): Are these heterogeneous processes driven by distinct molecular mechanisms?

3): How does the temporal coordination of these regulator drive the protrusion heterogeneity?

To answer these questions, we first acquired live cell imaging dataset tagged with different regulators and obtained more than 100 live cell videos perturbated by different drugs using PtK1 cells (rat kangaroo kidney epithelial cell line). The data from live cell imaging using fluorescence microscopy is particularly suited to deconvolve the subcellular protrusion heterogeneity because of its high temporal resolution compared to traditional imaging techniques such as static images acquired from immunofluorescence imaging. I propose machine learning-based pipelines to deconvolve the subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging and unravel the differential drug susceptibility of subcellular protrusion. This study is expected to provide a new framework to the cell migration research community.

A general methodology to study cell migration could be divided into four steps shown in Fig 1.2. The cells are engineered with chemical or fluorescent tag such as GFP (Green Florescence Protein) for the proteins of interest. Then, automated image acquisition is carried out by light microscopy techniques such as florescence microscopy or phase contrast microscopy. Various image processing techniques can be applied to image dataset to extract morphological and temporal features such as cell edge and velocity. Finally, extracted features can be further analyzed by advanced computational tools.

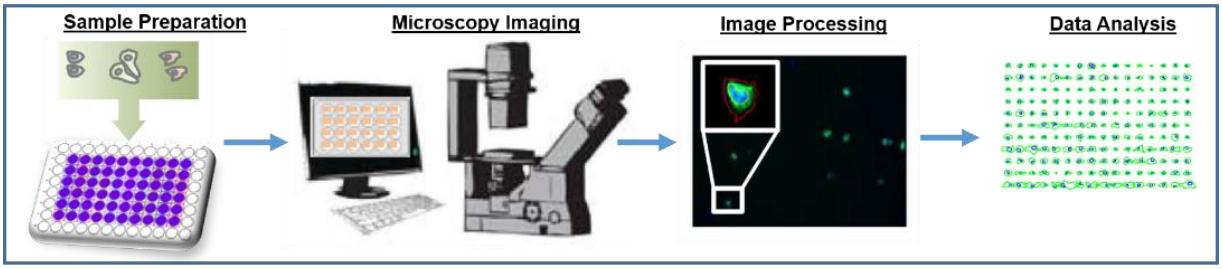


Fig. 1.2 Cell migration study methodology (modified from Wu 2015 [43])

We mainly focus on the last two steps after the live cell videos have already been acquired. Based on previous works[32], we propose advanced ML pipelines to identify distinct phenotypes with their actin regulator dynamics.

1.2 Machine Learning Review

1.2.1 Introduction

As a subfield of Artificial Intelligence, ML is a integrated research field with computer science, statistics and cognitive science[9-10]. The aim of ML is to develop computational tools to effectively perform specific tasks with minimal human instruction or intervention by learning tasks from dataset directly. Based on the availability of annotation (labeled data), we can divide ML into three categories: supervised learning and unsupervised learning and reinforcement learning. Supervised learning which requires the annotated dataset, discovers the relationship between inputs and outputs while unsupervised learning utilizes data representation as input and determine internal organization of data. Reinforcement learning as a popular topic in robotics mainly integrate the experience and current environment information to make an optimal action for further step to maximize future rewards (Fig. 1.3 (a)).

In current biological research, supervised learning and unsupervised learning are widely used[11-13]. In both supervised and unsupervised learning, two steps are involved: feature extraction and optimization including classification, regression or clustering (Fig. 1.3 (b)). The goal of feature extraction is to represent raw data using relevant features in a more effective way. The new representations derived from raw dataset have less noise, redundancy, and dimensionality, which are more informative and beneficial to subsequent computational processes. After representation or features are determined, supervised learning can be done for classification or regression or unsupervised learning for data clustering. The goal of these procedures is to optimize defined objective function related to criteria for specific tasks. For example, if the data are already annotated for supervised learning, we could train classifiers such as Support Vector Machine (SVM)[44], Random Forest[45] or Multilayer Perception (MLP) Neural network[46] using the annotated dataset. After training, we use the trained models to predict the outcome for new input data. For discovering hidden patterns within dataset, we can apply clustering methods such as k-mean[47], density-based clustering[48] and so on to divide whole dataset into separate groups related to hidden patterns.

In comparing to the second optimization step, feature extraction is highly dependent on the dataset. Most features are designed specifically based on the domain expert knowledge and difficult to apply to other fields without modification. Recently, Deep learning or representation learning[15-16] to learn effective features without human intervention revolutionized ML dramatically. Moreover, the representation extracted using pretrained models with large-scale datasets can be applied to other fields without modification, which benefits biological and biomedical areas since this could potentially reduce the require data size for effective training.

In this dissertation, we covered machine learning techniques from classification to clustering, from feature engineering to representation learning for cell imaging applications. In Chapter 2, we formulated the problem of the quantification of microbeads attached to cancer cells as a classification problem and utilized representation transfer learning technique to build an effective classifier. In Chapter 3, we used hand-crafted ACF-based features to deconvolute the subcellular protrusion heterogeneity, and in Chapter

4, we applied a deep learning method to learn features automatically for fine-grained deconvolution of subcellular heterogeneity.

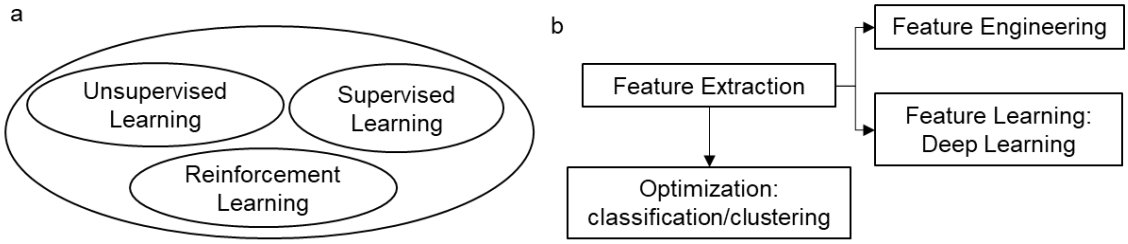


Fig 1. 3 Machine Learning Overview.

1.2.2 Time Series Clustering

Time series data or longitudinal data is ubiquitous in biological and biomedical fields[49-50] such as gene expression dataset, live cell movies[32,] and tumor growth measurement in drug discovery. We focus on cell protrusion phenomena represented by fluorescence time-lapse images, and extract protrusion velocity time series. There is no sufficient domain knowledge to annotate this dataset. Therefore, it is required to discover hidden patterns within this dataset by clustering.

Time series clustering[63] focuses on extract hidden patterns within time series dataset by measuring the similarity among time series. Many publications or review papers related to time series analysis are available[51-53]. Here, I reviewed the important techniques related to time series clustering.

1.2.2.1 Time Series Feature Extraction

Feature extraction techniques can be divided them into two categories: Manually-designed Features and Representation Learning. The purpose of feature extraction in time series clustering is to calculate the similarity between samples.

Manually-designed Features. In past decades, many features are designed specifically to time series data. The simple features for time series are various characteristics of time series such as trend, serial correlation, seasonality, periodicity and so on. Rob J Hyndman performed many works[54-55] to effectively extract those characteristics for time series analysis. The drawback of this method is that the features are highly depended on the datatype and quality of raw time series data. These feature extraction methods are effective with stationary time series data with low noise since the noise level would affect the quality of features. Therefore, denoising raw time series is necessary before feature extraction.

Another methodology to extract features from time series is to transform the raw time series into other domains. The well-known representation methods include Discrete Wavelet Transform (DWT)[56] and Discrete Fourier Transform (DFT)[57]. Also, other representation methods such as Piecewise Aggregate Approximation (PAA) and Symbolic Aggregate approXimation (SAX)[59] process time series data on time

domain were proposed. The main idea of PAA is to eliminate the local noise to maintain the global structure by local averaging. Furthermore, SAX transforms the PAA output into symbolic sequences. SAX and PAA are widely used to reduce the dimensionality in many different applications like acoustic signal analysis of bird vocalization[79], human motion analysis[80] and our time series protrusion velocity analysis[34]. In comparison to DWT and DFT, PAA and SAX have low computational complexity and simplicity with high sensitivity. Moreover, comparing with the time series data, there are much more image-related features designed in past years. Therefore, we can transform one-dimensional (1D) time series data into two-dimensional pseudo-image field[60]. After that, we can leverage image features for further time series analysis.

Additionally, motif discovery[61-62] is another popular method to distinguish different time series dataset, especially for variable-length time series dataset. The main idea of motif discovery is to discover overrepresented short time series patterns. The advantage is simple and data-dependent while the drawbacks is there is no guarantee that the discovered motifs have important meaning related to raw time series dataset.

Representation Learning. Representation learning is an attractive method to learn features with less human instruction. This includes two parts: time series modeling and deep learning. The main idea of representation learning is to utilize the parameters in the model as the features for further time series analysis.

Time series modeling[63] plays an important role in understanding the mechanism of complex system. Generally, the stochastic time series models could be divided into two parts: linear models including Autoregressive (AR), Moving Average (MA) and their variations and combinations like Autoregressive Moving Average (ARMA) and non-linear models like threshold AR[85]. The main idea of time series modeling is to simulate the time series patterns by mathematical equations like derivative functions. After fitting the model, we could utilize the fitted model for better prediction in future and the parameters of model to potentially represent the initial time series data. The limitation is that it still requires domain knowledge to select and build effective time series models.

Recently, autoencoder[64], a popular DL methodology, was proposed to learn feature from dataset directly. The main idea of autoencoder is that using the neural network structure to reconstruct the input data itself. If the neural network structure could successfully reconstruct input by minimizing the difference between input and reconstructed output, the hidden parameters inside the structure can have good representation for input data. Integrating the Long-Short Term Memory (LSTM)[65] or Gated Recurrent Units (GRU)[66], LSTM-autoencoder or GRU-autoencoder were proposed specifically to extend the autoencoder to time series data and had applied in many applications successfully. For example, in 2015, Srivastava proposed models based on LSTM-autoencoder that can learn good representations of video

sequences[81]. Further, Marchi[82] applied denoised autoencoder with bidirectional LSTM for acoustic novelty detection and achieved outstanding performance comparing with state-of-the-art methods. Moreover, Tang utilized LSTM-autoencoder to extract a mapping between acoustic and motion features for music-oriented dance choreography synthesis[83].

1.2.2.2 time series clustering methods

After feature extraction, there are many similarity measurements proposed for time series dataset like Dynamic Time Warping (DTW)[69-70]. Also, many R packages[67-68] are implemented for time series similarity measurements. Based on sample-sample similarity matrices, many clustering methods including K-means[47], Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[48], density peak[71], Self-Organizing Map (SOM)[72], community detection[73] can be applied. Regardless of clustering methods, it is also challenging to determine the optimal numbers of clusters for specific datasets since it is highly dependent on research questions. The typical methods to determine the cluster numbers are as follows: In the beginning, we recommend to estimate the optimal range of numbers of clusters using various external and internal criteria such as silhouette value. Then, we will obtain the clustering results with different numbers of clusters within the optimal ranges. Finally, using domain-knowledge and the purpose of study, we determine the optimal number of clusters and clustering results by merging different clusters or splitting some clusters into sub-clusters. This procedure is used in Chapter 3 and Chapter 4.

K-means[47] is a simple and widely-used method using raw data and similarity measurement method as input. The main idea of K-means is to estimate *K* centers to assign all the samples into nearest cluster represented by estimated centers based on the defined distance measurement. After determining the number of cluster *K*, an iterative refinement process would be used to update the estimated centers and the sample assignment by minimizing the within-cluster sum of squares until it's converged. Because the samples are always assigned to the nearest center estimated by ensemble average, K-means would be challenging to identify complex patterns like u-shape or star-shape clusters. In addition, K-means could not provide any information to suggest the optimal number of clusters, *K*.

Additionally, different from K-means, DBSCAN[48] as a most common clustering method, could successfully detect the complex patterns. DBSCAN locally connect the near samples together by the pre-defined radius of a neighborhood. Then, DBSCAN would classify all the samples into three categories: core samples if in the local neighborhood, many samples (more than a pre-specified threshold) are identified, reachable samples, which could be indirectly connected to the core samples and outliers, which is not reachable for any other samples. After that, DBSCAN partition all the samples into several clusters based on the reachability. One advantage of DBSCAN is to eliminate outliers based on the defined parameters to provide more purified clusters for further analysis. However, it is still a challenge to pre-define two effective parameters.

Density Peak clustering method was recently proposed[71] and used in our analysis because of its merits. First, Density Peak clustering is independent on the raw data space and only require the sample-sample similarity distance. Then, using only one pre-defined parameter, based on the distance, it would estimate the local density of each sample and relative distance from any points with a higher local density. After that, using the map visualized by local density and relative distance, it could suggest the optimal number of clusters in the data. Furthermore, density peak could trim the outliers and detect the complex pattern similar to DBSCAN. In our analysis[34], we applied Density Peak to successfully identify five distinguished subcellular protrusion phenotypes.

In addition, there are also several graph-based clustering widely-used, like SOM[72] and community clustering[73]. SOM, based on artificial neural network including training and mapping step, tries to map the high-dimensional feature to low-dimensional space. Since the training process is involved. The cluster results could be different for two runs and sometimes the samples from the same cluster could split into two different regions. community clustering was used in Chapter 4 for phenotypes identification because if its simplicity. Similar to Density Peaking clustering, community clustering is also independent on the raw data space and only distance matrix and one parameter to construct the nearest graph is required. The main idea of community clustering is that if there were a real community or cluster, containing many samples, random walk would be trapped in this community. Now community clustering is widely used in single-cell RNA analysis[84].

Besides the general clustering methods, I would briefly describe several clustering methods proposed specifically for time series data based on motif discovery. K-shape[74], as one of these methods, integrate normalized version of the cross-correlation measure as a distance measure and iterative refinement of cluster centroids from k-mean to divide the time series sample into different categories. Besides, shaplets[75] referring to time series subsequences which are maximally representative of a class in some sense are proposed for time series classification. After that, unsupervised shapelet (u-shapelet)[76], Unsupervised shapelet learning model (USLM)[77] and more derivative publications are proposed further for time series clustering.

1.3 Outline of dissertation

My dissertation is organized into three parts: In Chapter 2, we demonstrated that deep learning could extract useful features from diffraction patterns from lens-free digital in-line holography (LDIH) microscopy for molecular diagnostics for cancer cells[78]. In Chapter 3, we proposed the traditional ACF-based time series clustering pipeline called HACKS to deconvolute the subcellular protrusion heterogeneity[34]. In Chapter 4, we focused on the variable-length time series protrusion dataset and proposed a deep learning structure integrated with the outcome of HACKS for detailed and quantitative understandings of molecular mechanisms hidden in subcellular heterogeneity in pharmacologically and

metabolically perturbed epithelial cells. In Chapter 5, we summarized our contributions and proposed potential research directions based on our work.

Literature Cited

1. Abraham, Vivek C., D. Lansing Taylor, and Jeffrey R. Haskins. "High content screening applied to large-scale cell biology." *Trends in biotechnology* 22.1 (2004): 15-22.
2. Macarron, Ricardo, et al. "Impact of high-throughput screening in biomedical research." Nature reviews Drug discovery 10.3 (2011): 188.
3. Orth, Antony, et al. "Gigapixel multispectral microscopy." Optica 2.7 (2015): 654-662.
4. Orth, Antony, Diane Schaak, and Ethan Schonbrun. "Microscopy, meet big data." Cell systems 4.3 (2017): 260-261.
5. Marx, Vivien. "Biology: The big challenges of big data." (2013): 255.
6. Leonelli, Sabina. "Philosophy of Biology: The challenges of big data biology." *eLife* 8 (2019): e47381.
7. Altschuler, Steven J., and Lani F. Wu. "Cellular heterogeneity: do differences make a difference?." *Cell* 141.4 (2010): 559-563.
8. Erick Moen, Dylan Bannon, et al. “Deep learning for cellular image analysis” Nature method (2019): (https://doi.org/10.1038/s41592-019-0403-1)
9. Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006.
10. Hastie, Trevor, et al. "The elements of statistical learning: data mining, inference and prediction." The Mathematical Intelligencer 27.2 (2005): 83-85.
11. Kan, Andrey. "Machine learning applications in cell image analysis." *Immunology and cell biology* 95.6 (2017): 525-530.
12. Ovchinnikov, Sergey, et al. "Protein structure determination using metagenome sequence data." Science 355.6322 (2017): 294-298..
13. Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." *nature* 529.7587 (2016): 484..
14. Ng, Andrew. "Machine Learning and AI via Brain simulations." *Accessed: May* 3 (2013): 2018.
15. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.
16. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016
17. Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.
18. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
19. Shen, Dinggang, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." Annual review of biomedical engineering 19 (2017): 221-248.
20. Ching, Travers, et al. "Opportunities and obstacles for deep learning in biology and medicine." *Journal of The Royal Society Interface* 15.141 (2018): 20170387.
21. Webb, Sarah. "Deep learning for biology." *Nature* 554.7693 (2018).
22. Zitnik, Marinka, et al. "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities." *Information Fusion* 50 (2019): 71-91.
23. Sommer, Christoph, and Daniel W. Gerlich. "Machine learning in cell biology–teaching computers to recognize phenotypes." *J Cell Sci* 126.24 (2013): 5529-5539.
24. Xu, Wenbo, et al. "Digital in-line holography for biological applications." *Proceedings of the National Academy of Sciences* 98.20 (2001): 11301-11305.
25. Im, Hyungsoon, et al. "Digital diffraction analysis enables low-cost molecular diagnostics on a smartphone." *Proceedings of the National Academy of Sciences* 112.18 (2015): 5613-5618.

26. Lauffenburger, Douglas A., and Alan F. Horwitz. "Cell migration: a physically integrated molecular process." *cell*84.3 (1996): 359-369.
27. Guirguis, Raouf, et al. "Cytokine-induced pseudopodial protrusion is coupled to tumour cell migration." *Nature*329.6136 (1987): 261.
28. Mayor, Roberto, and Sandrine Etienne-Manneville. "The front and rear of collective cell migration." *Nature reviews Molecular cell biology* 17.2 (2016): 97.
29. Schumacher, Linus J., Philip K. Maini, and Ruth E. Baker. "Semblance of heterogeneity in collective cell migration." *Cell systems* 5.2 (2017): 119-127.
30. Shafqat-Abbasi, Hamdah, et al. "An analysis toolbox to explore mesenchymal migration heterogeneity reveals adaptive switching between distinct modes." Elife 5 (2016): e11384.
31. Gordonov, Simon, et al. "Time series modeling of live-cell shape dynamics for image-based phenotypic profiling." Integrative Biology 8.1 (2015): 73-90.
32. Lee, K. et al. Functional hierarchy of redundant actin assembly factors revealed by fine-grained registration of intrinsic image fluctuations. Cell Syst 1, 37-50, (2015).
33. Machacek, M., Hodgson, L., Welch, C., Elliott, H., Pertz, O., Nalbant, P., Abell, A., Johnson, G.L., Hahn, K.M., and Danuser, G. (2009). Coordination of Rho GTPase activities during cell protrusion. Nature 461, 99.
34. Wang, Chuangqi, et al. "Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging." Nature communications 9.1 (2018): 1688.
35. Krause, Matthias, and Alexis Gautreau. "Steering cell migration: lamellipodium dynamics and the regulation of directional persistence." *Nature reviews Molecular cell biology*15.9 (2014): 577.
36. Pollard, Thomas D., and John A. Cooper. "Actin, a central player in cell shape and movement." *Science* 326.5957 (2009): 1208-1212.
37. Pollard, Thomas D., and Gary G. Borisy. "Cellular motility driven by assembly and disassembly of actin filaments." *Cell*112.4 (2003): 453-465.
38. Ridley, Anne J., et al. "Cell migration: integrating signals from front to back." *Science* 302.5651 (2003): 1704-1709.
39. Block, Jennifer, et al. "FMNL2 drives actin-based protrusion and migration downstream of Cdc42." *Current Biology* 22.11 (2012): 1005-1012.
40. Rottner, K., Behrendt, B., Small, J. V. & Wehland, J. VASP dynamics during lamellipodia protrusion. *Nat. Cell. Biol.* **1**, 321–322 (1999).
41. Havrylenko, Svitlana, et al. "WAVE binds Ena/VASP for enhanced Arp2/3 complex–based actin assembly." *Molecular biology of the cell* 26.1 (2015): 55-65.
42. Siton-Mendelson, Orit, and Anne Bernheim-Groswasser. "Functional actin networks under construction: the cooperative action of actin nucleation and elongation factors." *Trends in biochemical sciences* 42.6 (2017): 414-430.
43. Wu, Pei-Hsun, et al. "Evolution of cellular morpho-phenotypes in cancer metastasis." *Scientific reports* 5 (2015): 18437.
44. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20.3 (1995): 273-297.
45. Ho, Tin Kam (1995). "Random Decision Forest". Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition, Montreal, Canada, August 14-18, 1995, 278-282
46. Rosenblatt, Frank. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. No. VG-1196-G-8. Cornell Aeronautical Lab Inc Buffalo NY, 1961.
47. MacQueen, James. "Some methods for classification and analysis of multivariate observations." *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. No. 14. 1967.
48. Kriegel, Hans-Peter, et al. "Density-based clustering." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.3 (2011): 231-240.

49. Bar-Joseph, Ziv. "Analyzing time series gene expression data." *Bioinformatics* 20.16 (2004): 2493-2503.
50. Enright, J. T. "The search for rhythmicity in biological time-series." *Journal of theoretical Biology* 8.3 (1965): 426-468.
51. Aghabozorgi, Saeed, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering–A decade review." *Information Systems* 53 (2015): 16-38.
52. Fu, Tak-chung. "A review on time series data mining." *Engineering Applications of Artificial Intelligence* 24.1 (2011): 164-181.
53. Keogh, Eamonn, et al. "Segmenting time series: A survey and novel approach." *Data mining in time series databases*. 2004. 1-21.
54. Wang, Smith and Hyndman (2006) Characteristic-based clustering for time series data. Data Mining and Knowledge Discovery, 13(3), 335-364.
55. Wang, Smith-Miles and Hyndman (2009) “Rule induction for forecasting method selection: meta-learning the characteristics of univariate time series”, Neurocomputing, 72, 2581-2594.
56. Shensa, Mark J. "The discrete wavelet transform: wedding the a trous and Mallat algorithms." *IEEE Transactions on signal processing* 40.10 (1992): 2464-2482.
57. Harris, Fredric J. "On the use of windows for harmonic analysis with the discrete Fourier transform." *Proceedings of the IEEE* 66.1 (1978): 51-83.
58. Lin, Jessica, et al. "Experiencing SAX: a novel symbolic representation of time series." *Data Mining and knowledge discovery* 15.2 (2007): 107-144.
59. Lin, Jessica, et al. "A symbolic representation of time series, with implications for streaming algorithms." *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. ACM, 2003.
60. Wang, Zhiguang, and Tim Oates. "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks." *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.
61. Chiu, Bill, Eamonn Keogh, and Stefano Lonardi. "Probabilistic discovery of time series motifs." *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003.
62. Mueen, Abdullah, et al. "Exact discovery of time series motifs." *Proceedings of the 2009 SIAM international conference on data mining*. Society for Industrial and Applied Mathematics, 2009.
63. Hamilton, James Douglas. *Time series analysis*. Vol. 2. Princeton, NJ: Princeton university press, 1994.
64. Ng, Andrew. "Sparse autoencoder." *CS294A Lecture notes*72.2011 (2011): 1-19.
65. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
66. Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).
67. Mori, Usue, Alexander Mendiburu, and Jose A. Lozano. "Distance measures for time series in R: The TSdist package." *R journal* 8.2 (2016): 451-459.
68. Montero, Pablo, and José A. Vilar. "TSclust: An R package for time series clustering." *Journal of Statistical Software* 62.1 (2014): 1-43.
69. Berndt, Donald J., and James Clifford. "Using dynamic time warping to find patterns in time series." *KDD workshop*. Vol. 10. No. 16. 1994.
70. Keogh, Eamonn, and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping." *Knowledge and information systems* 7.3 (2005): 358-386.
71. Rodriguez, Alex, and Alessandro Laio. "Clustering by fast search and find of density peaks." *Science* 344.6191 (2014): 1492-1496.
72. Vesanto, Juha, and Esa Alhoniemi. "Clustering of the self-organizing map." *IEEE Transactions on neural networks* 11.3 (2000): 586-600.

73. Pons, Pascal, and Matthieu Latapy. "Computing communities in large networks using random walks." *International symposium on computer and information sciences*. Springer, Berlin, Heidelberg, 2005.
74. Paparrizos, John, and Luis Gravano. "k-shape: Efficient and accurate clustering of time series." *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015.
75. Ye, Lexiang, and Eamonn Keogh. "Time series shapelets: a new primitive for data mining." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.
76. Ulanova, Liudmila, Nurjahan Begum, and Eamonn Keogh. "Scalable clustering of time series with u-shapelets." *Proceedings of the 2015 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2015.
77. Zhang, Qin, et al. "Unsupervised Feature Learning from Time Series." *IJCAI*. 2016.
78. Kim, Sung-Jin, et al. "Deep transfer learning-based hologram classification for molecular diagnostics." Scientific reports 8.1 (2018): 17003.
79. Kasten, Eric P., Philip K. McKinley, and Stuart H. Gage. "Automated ensemble extraction and analysis of acoustic data streams." *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*. IEEE, 2007.
80. Araki, Yutaka, et al. "Construction of symbolic representation from human motion information." International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, Berlin, Heidelberg, 2006.
81. Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhudinov. "Unsupervised learning of video representations using lstms." International conference on machine learning. 2015.
82. Marchi, Erik, et al. "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.
83. Tang, Taoran, Jia Jia, and Hanyang Mao. "Dance with Melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis." *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018.
84. Butler, Andrew, et al. "Integrating single-cell transcriptomic data across different conditions, technologies, and species." *Nature biotechnology* 36.5 (2018): 411.
85. Adhikari, Ratnadip, and Ramesh K. Agrawal. "An introductory study on time series modeling and forecasting." *arXiv preprint arXiv:1302.6613* (2013).

Chapter 2

Deep transfer learning-based hologram classification for molecular diagnostics

Lens-free digital in-line holography (LDIH) is a promising microscopic tool that overcomes several drawbacks (e.g., limited field of view) of traditional lens-based microcopy. However, extensive computation is required to reconstruct object images from the complex diffraction patterns produced by LDIH, which limits LDIH utility for point-of-care applications, particularly in resource limited settings. Here, we describe a deep transfer learning (DTL) based approach to process LDIH images in the context of cellular analyses. Specifically, we captured holograms of cells labeled with molecular-specific microbeads and trained neural networks to classify these holograms without reconstruction. Using raw holograms as input, the trained networks were able to classify individual cells according to the number of cell-bound microbeads. The DTL-based approach including a VGG19 pretrained network showed robust performance with experimental data. Combined with the developed DTL approach, LDIH could be realized as a low-cost, portable tool for point-of-care diagnostics.

All the listed results have been published in “Kim S J, **Wang C**, Zhao B, et al. Deep transfer learning-based hologram classification for molecular diagnostics[J]. Scientific reports, 2018, 8(1): 17003”. As the co-first author, I implemented the automatic detection of cell candidates, designed and trained the classifiers for multi-categorical classification, quantified the molecular profiling, compared the performance of DTL and CNN, and wrote the manuscript.

2.1 Introduction

Lens-free digital in-line holography (LDIH) is a powerful imaging platform that overcomes many of the limitations of traditional microscopy[1-6]. LDIH records diffraction patterns produced by samples, which can later be used to computationally reconstruct original object images. This strategy enables LDIH to image a large area (~mm2) while achieving a high spatial resolution (~μm). Furthermore, the simplistic optical design allows for compact setups, consisting of a semiconductor imager chip and a coherent light source. LDIH has been previously tested for potential point-of-care (POC) diagnoses[7]. Recently, we have advanced LDIH for the purpose of molecular diagnostics (D3, digital diffraction diagnostics)[3] wherein cancer cells were labeled with antibody-coated-microbeads, and bead-bound cells were counted for molecular profiling.

A major hurdle to translating LDIH into POC tests is the need for extensive computational power. In principle, diffraction patterns can be back-propagated to reconstruct human-friendly object images. The bottleneck lies in the recovery of phase information, lost during the imaging process. It has been shown that this information can be numerically recovered through iterative optimization[1, 8-13], but the process is

costly in computation time and requires high-end resources. To overcome this issue, it was demonstrated that a deep neural network could be trained to recover phase information and reconstruct object images, substantially reducing the total computational time[14]. However, this method still required an input of back-propagation images obtained from the holograms. In this paper, we explored an alternative approach in which diagnostic information could be extracted from the raw hologram images without the need for hologram reconstruction. In the microbead-based assay, we reasoned that cell-bead objects could generate distinct hologram patterns, albeit imperceptible to human eyes, recognizable by machine vision classifiers. Developing such a capacity would eliminate the need for image reconstruction, further advancing LDIH utility for POC operations.

We here report on new machine-learning (ML) based approaches for LDIH image analysis. ML has been making significant progress in extracting information from complex biomedical images and started to outperform human experts for many data sets[15-18]. In this paper, we took deep transfer learning (DTL)[19-25] approach to classify raw holograms and compared them with other ML schemes including convolutional neural networks (CNN)[26-28]. DTL extracts feature information from input data using the convolution part of pre-trained networks and subsequently feeds the information to classifiers. It has been known that pretrained networks can be exploited as a general-purpose feature extractor[20]. In this DTL approach, we used a VGG19 model[29] that was pretrained with a large number of ordinary images (i.e., not holograms) available in the ImageNet[30] and fine-tuned the classifier to obtain high-performance classification. We applied these approaches to classifying holograms generated from cells and microbeads without a reconstruction process. Specifically, algorithms were developed to i) automatically detect the holograms of cells labeled with microbeads, ii) classify detected cells according to the number of the cell-bound beads, and iii) construct the histogram of the cell-bound beads from the entire hologram. We found that a DTL approach offered more reliable, robust, and efficient performance in hologram classification than the conventional CNN.

2.2 Materials and Methods

2.2.1 Data Collection

Samples were prepared by labeling cancer cells (SkBr3, A431) with polystyrene beads (diameter, 6 μm). We prepared four different sets of beads. Three sets were conjugated with antibodies against different molecular targets: EGFR, EpCAM, and HER2; the fourth set was conjugated with control IgG antibodies. Aliquots of cells were labeled with each set of the bead. Labeled cells, suspended in buffer, were loaded on a microscope slide, and their holograms were imaged using LDIH system[3]. To prepare the dataset set for classification, we reconstructed object images from holograms using a previously developed algorithm[3]. We cropped holograms (270 × 270 pixels) around the position of the automatically detected cell candidates (see Cell Candidate Detection below). Three researchers manually annotated the holograms of the cropped cell candidates with the following labels: the numbers (0, 1, 2, 3, ≥4) of the

beads attached to cells, the beads unattached to cells, multiple cells, and artifacts. Later, we collectively labeled the beads unattached to cells, multiple cells, and artifacts as 'background.'

2.2.2 Cell Candidates Detection

We implemented computational methods which automatically localized the single-cell candidates in the hologram image based on the diffracted patterns of concentric circles in holograms[32]. The algorithm uses the fact that the gradient directions of holograms on concentric circles converge to the centers of the diffraction patterns. The detailed detection procedure is the following:

1. We normalized the holograms by dividing the pixel values by background and then rescaled them into a range [0, 255].
2. We denoised the normalized hologram using Gaussian blurring with 6-pixel size (MATLAB function *imgaussfilt()*). Then we calculated the gradient direction and magnitude of the denoised holograms using the MATLAB build-in function *imgradient()* with '*perwitt*' method.
3. We thresholded the gradient magnitude images using the threshold value 8.0, which removed the small gradient magnitude pixels and generated the binary mask. Then, the gradient direction images were masked by the gradient magnitude binary mask.
4. Along each direction in the masked gradient direction images, the frequencies of the gradient directions were accumulated within a specified range (50-pixel length, which generated the frequency maps of the gradient direction.
5. We denoised the frequency accumulation map using Gaussian blurring with 3-pixel size (MATLAB function *imgaussfilt()*). Then, we thresholded the denoised frequency accumulation map using the top 1% of the pixel values and locate the center candidates. Then, we cropped 270 × 270 hologram and object image patches around the detected candidate center positions.

After detection, for each candidate, three annotators independently labeled cropped holograms and their corresponding object images. In order to balance the class distribution, we augmented the image data labeled with NB = 1, 2, 3, and $\geqslant$4. The augmentation was performed by two strategies: rotation with a range of [0, 40] and zooming-in with the maximum value, 0.2 using Keras library.

2.2.3 Machine Learning Classification

Using VGG19 pretrained model, we extracted image features from cropped holograms and object images. Since VGG19 was originally used for color images (RGB channels), and our data in a gray scale were duplicated to each channel. For data preprocessing, we perform the standard normalization, where each image patch was subtracted by its mean value and divided by its standard deviation. After the features extracted from VGG19, PCA (Principal Component Analysis) was performed to reduce the dimensionality of the data from 32768 to 500.

After the feature extraction step, we used an MLP (Multilayer Perceptron) consisting of three fully-connected neural network blocks for the classification. The first two blocks have a fully-connected (FC) layer, Batch Normalization layer, ReLU activation and Dropout layer (parameter: 0.5). The FC layers in the first two blocks have the sizes of 128 and 64, and the L2 norm regularizer (parameter: 0.05). The third block has an FC layer with 'softmax' activation. Also, Support Vector Machine (SVM) and Random Forest (RF) were applied to compare the performance with the MLP. The parameters of SVM and RF were optimized by the grid-search method in *sklearn* package shown in Table 2.1 and Table 2.2.

Table 2.1 The parameters of the grid search for RF

	n_estimators	max_features	max_depth
RF in binary classification	20, 50, 100	'auto', 'sqrt'	[none, 2, 20, 50]
RF in 5 categories classification	20, 50, 100, 500	'auto', 'sqrt'	[none, 2, 20, 50, 100]

Table 2.2 The parameters of the grid search for SVM

	Kernel	C	gamma
SVC	'RBF', 'Linear'	0.1, 1, 10, 100	0.1, 0.01, 0.001

To show the roles of the pretrained VGG19, we trained a CNN convolution neural using the same dataset (Fig. 7A). The CNN has three feature extraction blocks consisting of two convolutions layers and one max-pool layer (4 × 4) After this feature extraction, the same MLP structure was used for the classification.

2.2.4 Performance evaluation of the classifiers

We split the augmented dataset into three groups in a stratified fashion using the class labels: training, validation and test sets (64:16:20). The training set (64%) was used for training the network. The validation set (16%) was used for the model selections. After the training, the classification performance was evaluated using the testing set (20%). For the robust statistical analysis, we repeated the training 20 times. The performance measures were the accuracy, Cohen's Kappa coefficient (Kappa)[34], relative classifier information (RCI)[36]. Kappa is a standard metric for a multi-categorical classification and RCI, as an entropy-based measure, is also suitable to evaluate the performance by measure the reduced uncertainty by the classifier in comparison to the prior class distribution. For the classification involved with negative and positive bead attachment, we also measure sensitivity and specificity using *sklearn.metrics* python package. The samples of the positive bead attachment were treated as 'positive,' and the other cases were treated as 'negative'. For the statistical testing, we used unpaired two-tailed Wilcoxon rank sum test method which does not rely on the assumption of the Gaussian distribution.

2.2.5 Molecular Profiling

To quantify the distribution of the proportion or the frequency of the number of the attached beads (NB), we chose 18 images, whose cell candidates were larger than 15. For each image, we calculated the proportion and the frequency of the predicted and the actual numbers of the attached beads in each hologram.

Comparison between VGG19-PCA-MLP and CNN. To evaluate the performance between VGG19-PCA-MLP and CNN, the performance measures including accuracy, Kappa and RCI were used as described above. Then, the fluctuations of the validation accuracy and loss were measured as follows: we selected the last 20 epochs for each training process, and then calculate the residuals by subtracting the sample mean value. We repeated the training twenty times with random data splitting. The statistical test for the difference of variance was performed by two sample F-test.

2.3 Results

2.3.1 Systems and assay setup

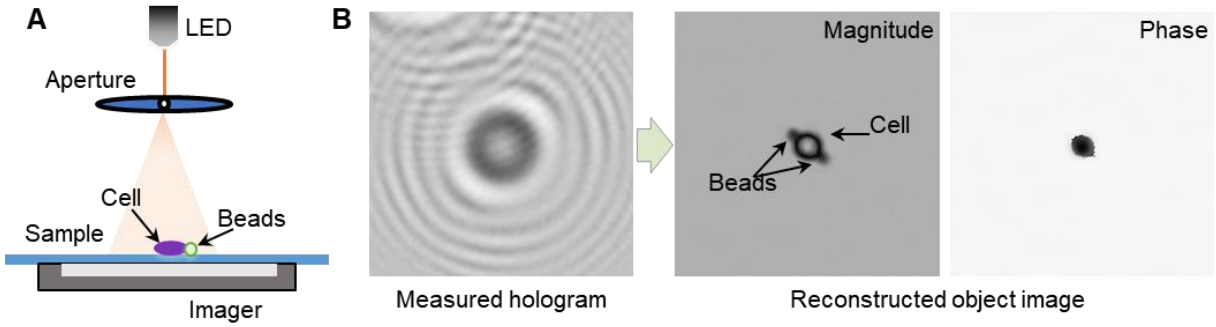


Fig. 2.1 In-line holographic imaging**.** (A) A holography system includes LED, a sample glass and a sensor where light is passed to a sample through a pinhole disk. (B) A hologram image and its associated reconstructed images consisting of magnitude and phase images.

The schematic of LDIH system[3] is shown in Figure 2.1A. As a light source, we used a light-emitting diode (LED: λ = 420 nm). The light passes through a circular aperture (diameter, 100 μm), generating a coherent spherical wave on the sample plane. The incidence light and the scattered light from the sample interfere with each other to generate holograms which are then recorded by a CMOS imager[10,31]. The system has a unit (×1) optical magnification, resulting in a field-of-view equal to the imager size.

To enable molecular-specific cell detection, we used antibody-coated microbeads (diameter, 6 μm) for cell labeling. The number of attached beads is proportional to the expression level of a target marker, allowing for quantitative molecular profiling3. Diffraction patterns from unlabeled and bead-bound cells have subtle differences that are hard to detect with human eyes (Fig. 2.1B). Only after image reconstruction can beads and cells be differentiated and counted; cells have high amplitude and phase values, whereas microbeads have negligible phase values.

2.3.2 Reconstruction-free ML approaches

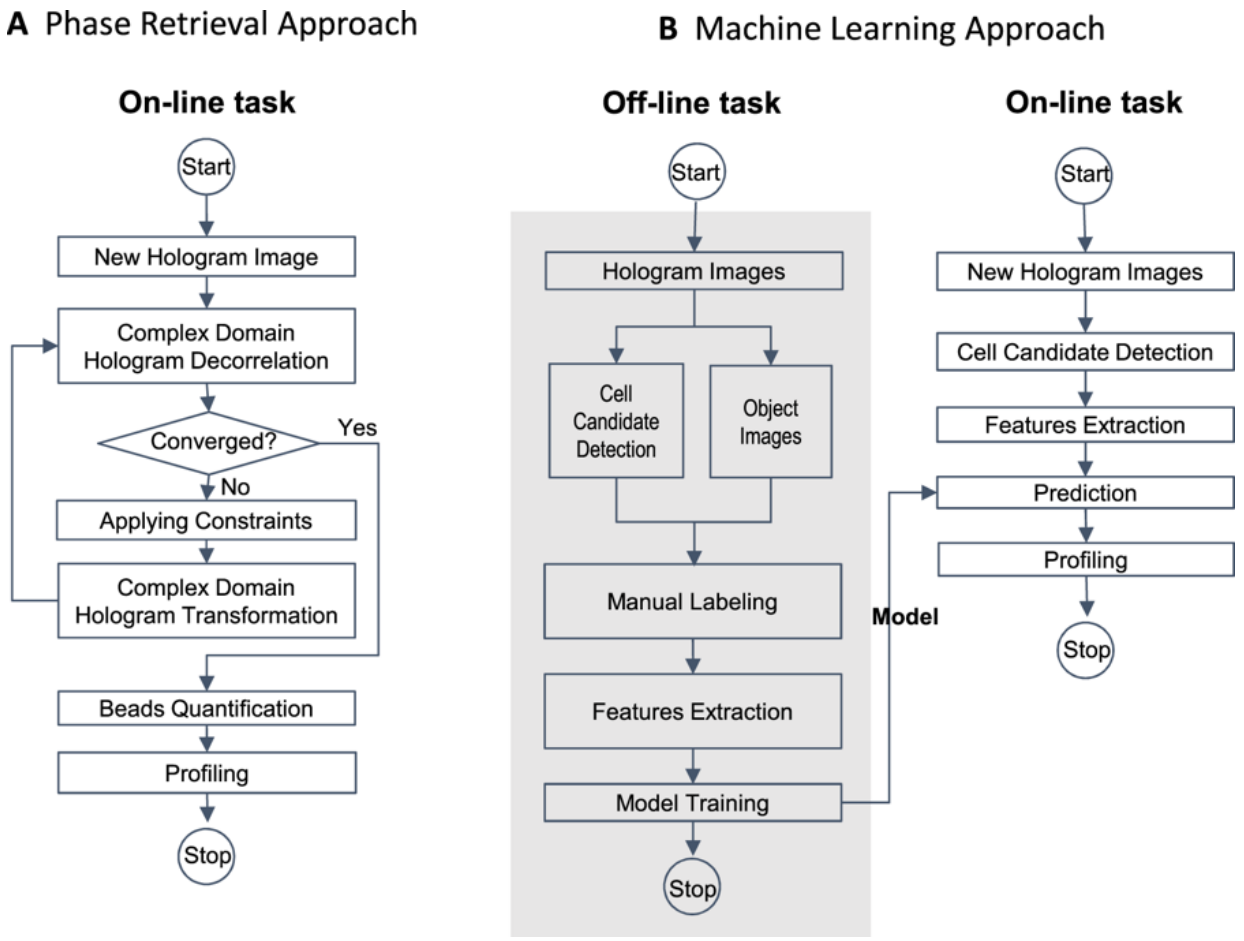


Figure 2.2 Flow charts of holographic diagnostic approaches**.** (A) A conventional approach includes iterative reconstruction processes by phase retrieval. (B) A machine learning based workflow for hologram classification.

Conventional LDIH reconstruction (Fig. 2.2A) requires multiple repetitions of back-propagation, constraint application, and transformation[8]. This iterative algorithm is computationally intensive, either incurring long processing time or requiring high-end resources (e.g., a high-performance graphical processing unit server) for faster results[3]. Furthermore, human curation is occasionally needed to correct for stray reconstruction (e.g., debris, twin images). In contrast, our ML-based approach is a reconstruction-free classification method (Fig. 2.2B). As an off-line task, we first build a training dataset by automatically detecting cell candidates and cropping them from the entire holograms. Then, we labeled each cropped hologram with the number of the cell-bound beads using reconstructed image as ground truth. Next, we trained a network using annotated holograms of bead-bound cells. After the training was complete, the network was used for online classification tasks; cell candidate holograms were detected and their holograms, without any image preprocessing, were entered as input for classification based on the number of cell bound beads. Finally, the histograms of the cell-bound beads from the entire holograms were created for molecular diagnosis.

Both off-line and on-line tasks in the ML approach (Fig. 2.2B) required the automatic detection of holographic patterns of cells. To achieve this task, we implemented a computational method which identifies the center of individual diffraction patterns[32]. First, the images of the gradient magnitude of holograms were generated and thresholded based on their intensity. The converging directions of gradients were used to estimate the positions of cell candidates in holograms (Fig. 2.3A; see Methods). Using this method, we detected 2729 potential cell candidates from 31 holograms. The samples for these holograms were prepared by labeling SkBr3 breast carcinoma cells with polystyrene beads conjugated with control, EpCAM, and HER2 antibodies. Then, we reconstructed object images and cropped the holograms and the object images (270 × 270 pixels). We labeled the cropped holograms (Fig. 2.3C) and their reconstructed object images (Fig. 2.3D) with the number of the bead attached to a cell (NB: 0, 1, 2, 3, $\geqslant$ 4). There were also images of floating beads, multiple cells, and artifacts, which were collectively labeled as 'background' (BG). The distribution of the class in the final training set is shown in Fig. 2.3B.

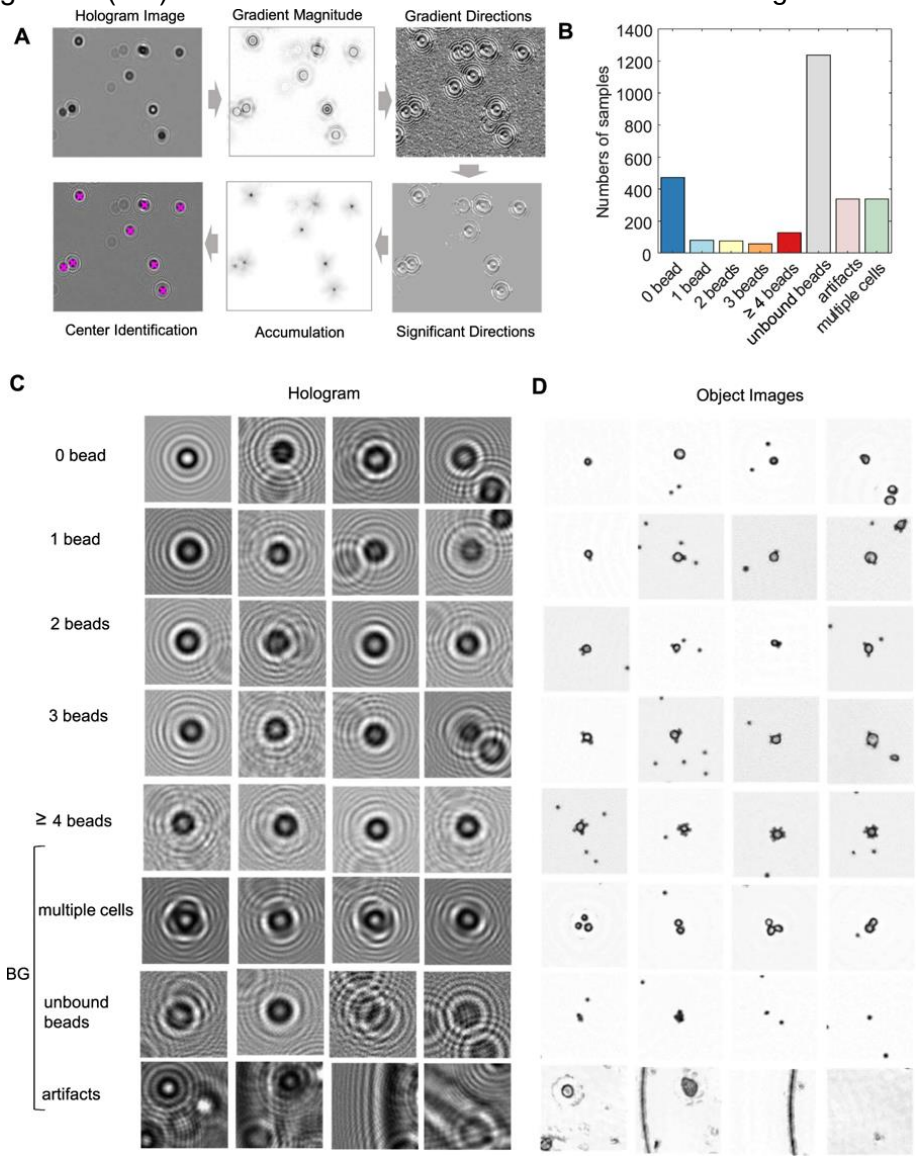


Fig. 2.3 Training set preparation for hologram classification**.** (A) Cell candidates detection workflow. (B) Class distribution in the training dataset. (C,D) Sample examples of holograms (C) and corresponding object images (D).

2.3.3 Visualization of hologram features

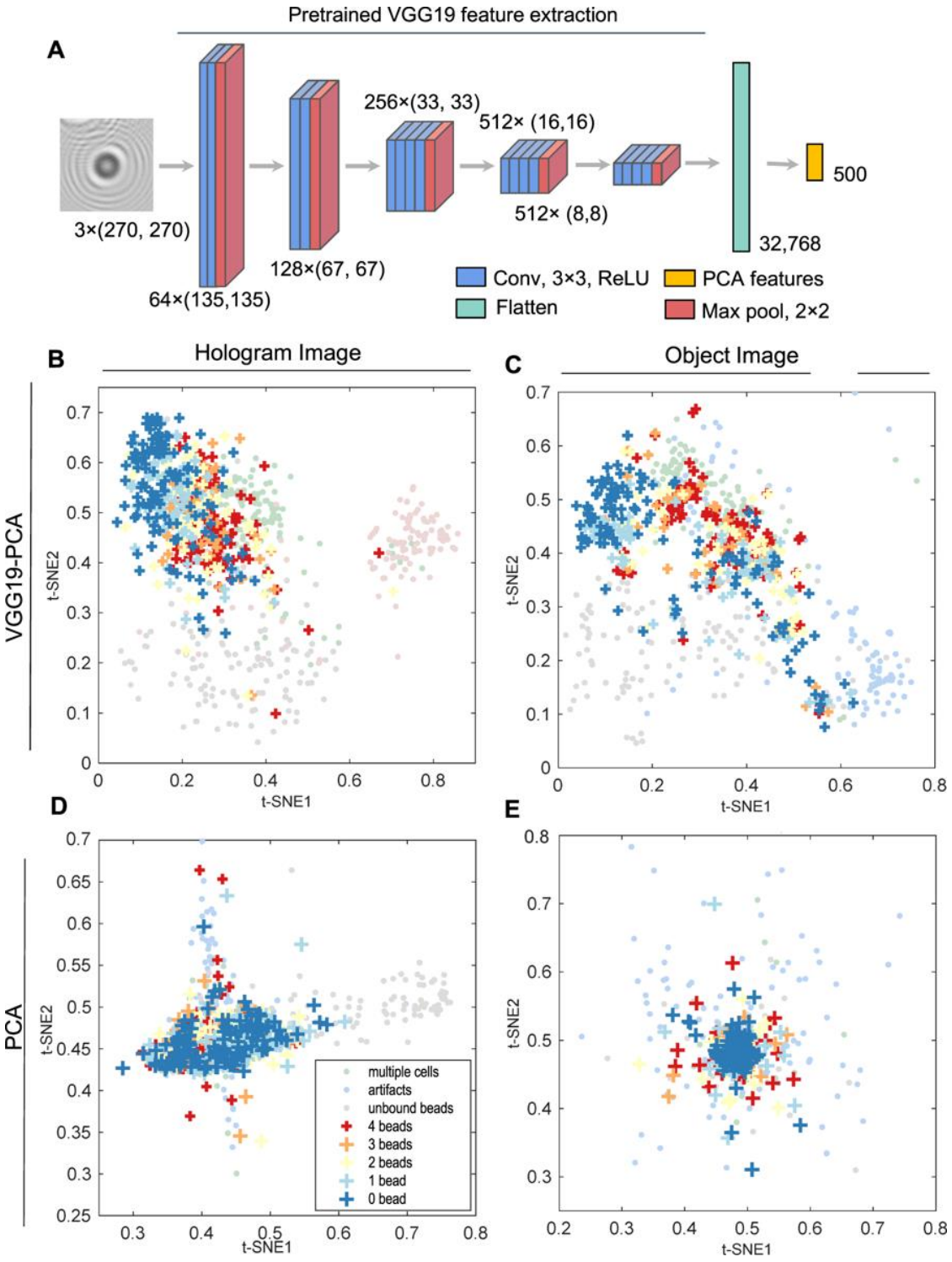


Fig. 2.4 Feature extraction from holograms**.** (A) Features extraction by the pretrained VGG19 model and PCA. (B–D) t-SNE plots of the extracted features. VGG19-PCA feature extraction from the holograms (B) and object images (C). PCA feature extraction from the holograms (D) and object images (E).

We first tested the feasibility of the reconstruction-free classification by visualizing the features extracted from the holograms. Using VGG19 pretrained model, we extracted features from the training set of

holograms (Fig. 2.4A). Since VGG19 was trained using color images (RGB channels) and our data were in a gray scale, we copied the same image in each channel in the VGG-19 pretrained model. Then, using PCA (Principal Component Analysis), we reduced the feature dimension from 32,768 to 500 and visualized their distribution using t-SNE plots[33]. In both holograms (Fig. 2.4B) and object images (Fig. 2.4C), each class of bead-bound cells was visually more segregated than the cases where only the same PCA was applied without using VGG19 (Fig. 2.4D-E), suggesting that VGG19-based features of the holograms could discriminate the numbers of cell-bound beads.

2.3.4 Classification results by deep transfer learning

Using the features from VGG19-PCA or PCA, we trained the multilayer perceptron (MLP, Fig. 2.5A) separately for holograms or object images. Since the training data were unbalanced (Fig. 2.3B), we took the following approaches: To balance the training set of cell-bound beads, we applied the data augmentation (rotation and zoom-in) to increase the data size in the case of NB $\geqslant$ 1 (see Methods). Then, to address the unbalance between bead-bound cells (NB : 0 to $\geqslant$ 4) and background (BG) data we used the weighted cost function using the proportion of bead-bound cells to BG data. From the whole dataset consisting of 2729 cropped images from 31 holograms, we randomly split the data into training, validation, and testing dataset with a 64:16:20 ratio. The model was selected based on the validation loss, and the model performance was evaluated based on the testing data. For statistical analysis, we repeated the training 20 times with different data splitting (see Methods for detail).

Since cells with more than two bead attachments are considered positive for a given target biomarker[3], we first performed the binary classification (N/P) based on the bead number (NB): negative (NB$\leqslant$ 1) vs. positive (NB$\geqslant$ 2). The accuracies of VGG19-PCA-MLP in N/P were 90.2% for holograms and 93.4% for object images, whereas the accuracies of PCA-MLP were only 79.5% for holograms and 76.8% for object images (Fig. 2.5B and Table 2.3). Since the background (BG) data are included in real situations, we also trained the classifiers after adding the BG class (N/P+BG). In comparison to the N/P classification, the accuracies were still similar (VGG19-PCA-MLP: 90.4% for holograms and 92.3% for object images, PCA-MLP: 79.1% for holograms and 76.4% for object images) (Fig. 2.5B and Table 2.3). The sensitivity and specificity also showed that VGG19-PCA-MLP outperformed PCA-MLP in all cases (Fig. 2.5C, Table 2.4 and 2.5).

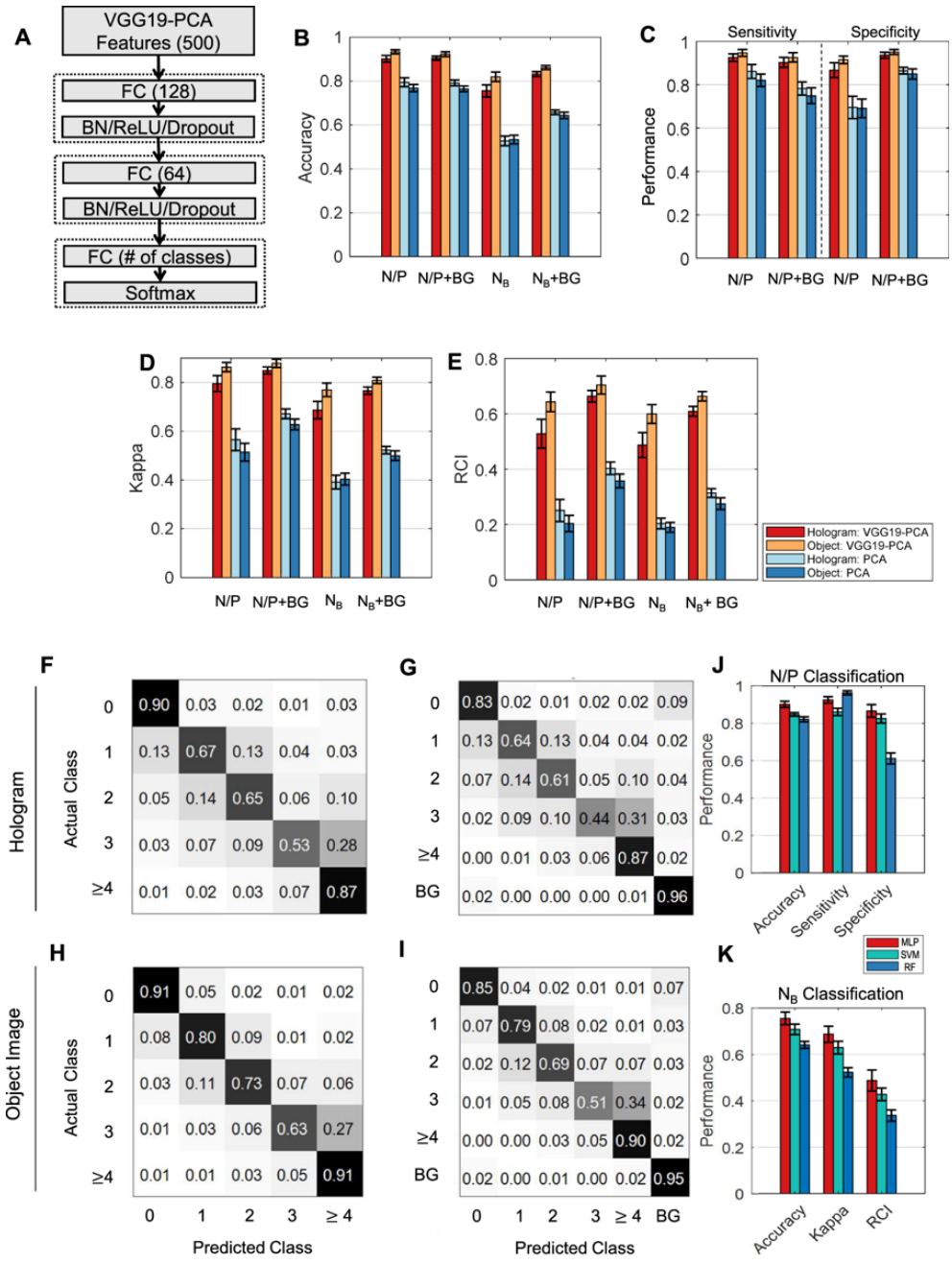


Fig. 2.5 Classification performance of the deep transfer learning for holograms**.** (A) MLP (Multi-Layer Perceptron) neural network classifier used in this study (FC: fully connected layer, BN: batch normalization layer). (B–E) Performance Comparison between VGG19-PCA-MLP and PCA-MLP. N/P: negative (NB ⩽ 1) and positive (NB ⩾ 2) bead attachment. N/P + BG: NB ⩽ 1, NB ⩾2, BG (background class), NB: the numbers of beads (0, 1, 2, 3, ⩾ 4 beads), NB + BG: the numbers of beads (0, 1, 2, 3, ⩾4), BG. The performance measures are accuracy (B), sensitivity/specificity (C), Cohen's Kappa (D), and RCI (E). (F–K) Average confusion matrices using VGG19-PCA-MLP using NB (five classes) classification (F) and NB + BG (6 classes) classification (G) for holograms, and NB classification (H), NB + BG classification (I) for object images. (J, K) The performance comparisons of VGG19-PCA-MLP (MLP), Support Vector Machine (SVM), Random Forest (RF) in N/P (two classes) classification (J) and NB (five classes) classification (K) using holograms.

Table 2.3 Accuracies of VGG19-PCA-MLP and PCA-MLP classifiers. The values within parentheses are the standard deviations. The p-values testing the difference between PCA-MLP and VGG19-PCA-MLP were obtained by Wilcoxon rank sum test.

Classification Type	Holograms			Object Image		
	PCA-MLP	VGG19-PCA-MLP	P-values	PCA-MLP	VGG19-PCA-MLP	P-values
N/P	0.795 (0.020)	0.902 (0.015)	6.625×10^{-08}	0.768 (0.017)	0.934 (0.009)	6.569×10^{-08}
N/P + BG	0.791 (0.013)	0.904 (0.009)	6.70×10^{-08}	0.764 (0.013)	0.923 (0.011)	6.644×10^{-08}
N_B	0.528 (0.022)	0.755 (0.027)	6.729×10^{-08}	0.534 (0.019)	0.820 (0.021)	6.70×10^{-08}
N_B + BG	0.658 (0.010)	0.832 (0.011)	6.653×10^{-08}	0.643 (0.014)	0.862 (0.009)	6.625×10^{-08}

Table 2.4 Sensitivities of VGG19-PCA-MLP and PCA-MLP classifiers. The values within parentheses are the standard deviations. The p-values testing the difference between PCA-MLP and VGG19-PCA-MLP were obtained by Wilcoxon rank sum test.

Classification Type	Hologram Image			Object Image		
	PCA-MLP	VGG19-PCA-MLP	P-values	PCA-MLP	VGG19-PCA-MLP	P-values
N/P	0.861 (0.031)	0.925 (0.017)	1.020×10^{-07}	0.820 (0.027)	0.947 (0.015)	6.523×10^{-08}
N/P + BG	0.782 (0.029)	0.902 (0.022)	6.644×10^{-08}	0.749 (0.036)	0.926 (0.021)	6.560×10^{-08}

Table 2.5 Specificities of VGG19-PCA-MLP and PCA-MLP classifiers. The values within parentheses are the standard deviations. The p-values testing the difference between PCA-MLP and VGG19-PCA-MLP were obtained by Wilcoxon rank sum test.

Classification Type	Hologram Image			Object Image		
	PCA-MLP	VGG19-PCA-MLP	P-values	PCA-MLP	VGG19-PCA-MLP	P-values
N/P	0.695 (0.051)	0.867 (0.034)	$8.256x\ 10^{-08}$	0.690 (0.043)	0.915 (0.017)	$6.304x\ 10^{-08}$
N/P + BG	0.865 (0.016)	0.936 (0.014)	$6.682x\ 10^{-08}$	0.850 (0.023)	0.951 (0.011)	$6.504x\ 10^{-08}$

While this binary classification for the negative and positive cells can be applied to molecular diagnostics, the actual number of the beads and their distribution from a given patient sample can provide more detailed information including cancer stages and patient sub-types. Therefore, we trained the classifiers

based on the numbers of the cell-bound beads. When we performed the classification using the number of beads (0, 1, 2, 3, $\geqslant$ 4), VGG19-PCA-MLP achieved significantly higher accuracies, 75.5% for holograms and 82.0% for object images than PCAMLP (52.8% for holograms and 53.4% for object images) (Fig. 2.5B and Table 2.3). When BG class was considered together for the real application (NB+BG), VGG19-PCAMLP achieved 83.2% for holograms, and 86.2% for object images, whereas PCA-MLP achieved 65.8% for holograms and 64.3% for object images (Fig. 2.5B and Table 2.3). The distinctiveness of the BG class from the other classes (Fig. 2.4B-C) increased the overall classification accuracies.

To quantitatively compare the classification performance among all classification cases, we employed the Cohen's kappa coefficient[34] and the relative classifier information (RCI)[35,36] (Fig. 2.5D-E, Table 2.6 and 2.7). Cohen's kappa compensates for classifications by random chance, and RCI quantifies how much uncertainty had been reduced by the classification relative to the prior probabilities of each class. Both measures are between 0 and 1 (0: worst, 1: perfect classification). The NB classification using VGG19-PCA-MLP and the holograms produced the significantly larger values of Cohen's kappa (0.687) and RCI (0.487) than the PCA-MLP (Cohen's kappa: 0.392, RCI, 0. 0.204, Table 2.6 and 2.7). The results showed the followings; i) N/P classifiers has better performance than NB classifiers since multicategory classification is more prone to error than binary classification. ii) The VGG19-PCAMLP outperforms PCA-MLP in all cases, iii) While the classification using holograms showed good performance, the classification using object images performs marginally better than holograms.

Table 2.6 Cohen's Kappa of VGG19-PCA-MLP and PCA-MLP classifiers. The values within parentheses are the standard deviations. The p-values testing the difference between PCA-MLP and VGG19-PCA-MLP were obtained by Wilcoxon rank sum test (All pvalues are the same since they have the same rank order distribution and there is no overlap of the data distribution of VGG19-PCA-MLP and PCA-MLP)

Classification Type	Hologram Image			Object Image		
	PCA-MLP	VGG19-PCA-MLP	P-values	PCA-MLP	VGG19-PCA-MLP	P-values
N/P	0.565 (0.043)	0.795 (0.032)	6.786×10^{-08}	0.514 (0.035)	0.863 (0.019)	6.796×10^{-08}
N/P + BG	0.672 (0.019)	0.849 (0.014)	6.796×10^{-08}	0.628 (0.021)	0.879 (0.017)	6.796×10^{-08}
N_B	0.392 (0.028)	0.687 (0.034)	6.796×10^{-08}	0.403 (0.024)	0.770 (0.027)	6.796×10^{-08}
N_B + BG	0.522 (0.014)	0.766 (0.015)	6.796×10^{-08}	0.500 (0.019)	0.808 (0.013)	6.796×10^{-08}

To see the performance of the multi-category classification more closely, we computed the average confusion matrices of the bead classification (NB, NB+BG). The prediction accuracy was high when the number of beads is 0 or $\geqslant$4, or the BG class, and the accuracies decreased when the bead number was between 1 and 3 (Fig. 2.5F-I). Since the high occurrence values of the confusion matrices were near the diagonals, the misclassification mainly happened among neighboring numbers for both holograms and object images. This property makes the molecular profiling from the entire holograms less susceptible to mis-classification error.

To compare the performances of different classifiers in our DTL approach, we also trained SVM (Support Vector Machine) and RF (Random Forest) using the same VGG19-PCA features. In both N/P and NB classifications, MLP outperformed SVM and RF significantly (see p-values in Table 2.8 and 2.9).

Table 2.7 RCI of VGG19-PCA-MLP and PCA-MLP classifiers. The values within parentheses are the standard deviations. The p-values testing the difference between PCA-MLP and VGG19-PCA-MLP were obtained by Wilcoxon rank sum test (All pvalues are the same since they have the same rank order distribution and there is no overlap of the data distribution of VGG19-PCA-MLP and PCA-MLP)

Classification Type	Hologram Image			Object Image		
	PCA-MLP	VGG19-PCA-MLP	P-values	PCA-MLP	VGG19-PCA-MLP	P-values
N/P	0.251 (0.040)	0.528 (0.052)	6.786×10^{-08}	0.204 (0.030)	0.643 (0.036)	6.796×10^{-08}
N/P + BG	0.403 (0.023)	0.664 (0.021)	6.796×10^{-08}	0.358 (0.025)	0.704 (0.033)	6.796×10^{-08}
N_B	0.204 (0.020)	0.487 (0.045)	6.796×10^{-08}	0.190 (0.018)	0.599 (0.034)	6.796×10^{-08}
N_B + BG	0.314 (0.015)	0.609 (0.018)	6.796×10^{-08}	0.275 (0.021)	0.664 (0.017)	6.796×10^{-08}

Table 2.8 P-values of the hypothesis testing of the differences in the performance measures among MLP, SVM, RF in N/P classification. Wilcoxon rank-sum test was used.

P-values	MLP VS SVC	MLP VS RF	SVC VS RF
Accuracy	6.532×10^{-08}	6.588×10^{-08}	4.356×10^{-06}
Sensitivity	9.197×10^{-08}	6.279×10^{-07}	6.188×10^{-08}
Specificity	0.0001054	6.560×10^{-08}	6.504×10^{-08}

Table 2.9 P-values of the hypothesis testing of the differences in the performance measures among MLP, SVM, and RF in NB classification. Wilcoxon rank-sum test was used.

P-values	MLP VS SVC	MLP VS RF	SVC VS RF
Accuracy	1.094×10^{-05}	6.710×10^{-08}	7.184×10^{-08}
Kappa	1.415×10^{-05}	6.796×10^{-08}	6.796×10^{-08}
RCI	2.925×10^{-05}	6.796×10^{-08}	6.796×10^{-08}

2.3.5 Molecular profiling using the deep transfer learning

When it comes to the molecular diagnosis using LDIH, the clinical decision is often made at the cell population level. Therefore, we assessed how our hologram multi-category classification (NB) matched with the distribution of the cell-bound beads from an entire hologram. We overlaid the actual and predicted distributions of cell-bound beads from 18 different samples, whose number of detected cell candidates are more than 15 excluding BG (Fig. 2.6A). We also plotted that the histograms of the actual and predicted numbers of the cell-bound beads in each sample (Fig. 2.6B). These show that the predicted bead proportions matched well with the actual distribution. Also, the mean difference between the proportions of the actual and the prediction was within 5% (Fig. 2.6C). This suggests that our multi-category classification based on the number of the cell-bound beads can be used to characterize the molecular profiles of the cancer cell population from a patient sample.

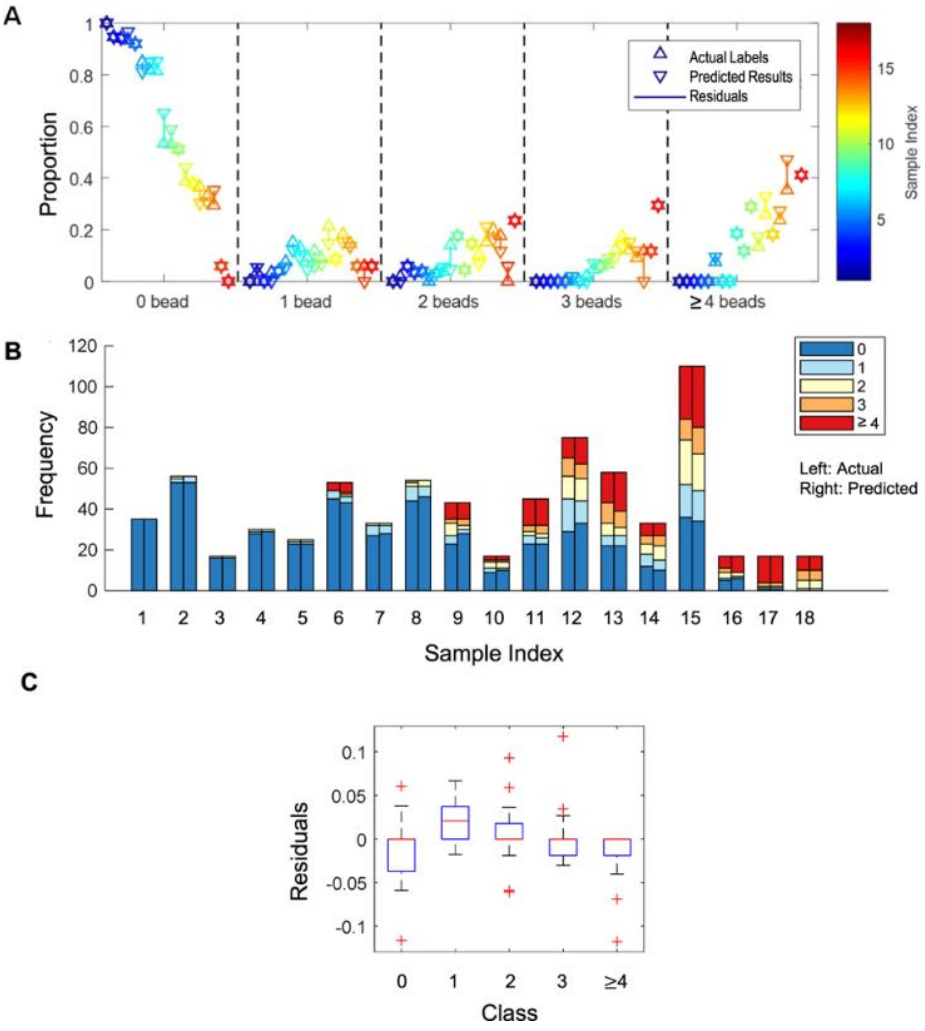


Fig. 2.6 Molecular profiling using deep transfer learning**.** (A, B) Comparison of the proportions (A) and numbers (B) of the cell-bound beads in each entire hologram between the actual labels (left) and predicted results (right). The classifier was built without BG class (NB). The color represents each hologram. (C) The residuals between actual and predicted proportions in each class.

2.3.6 Roles of VGG19 pretrained model

To evaluate the role of the pretrained model in our classification, we also trained a conventional convolutional neural network (CNN) de novo for NB classification (Fig. 2.7A). Whereas there were no statistical differences in accuracy, RCI, and kappa between the CNN and the VGG19-PCAMLP (Fig. 2.7B, Table 2.10), we observed that the validation loss and accuracy of the training curves of this CNN were far more fluctuating than VGG19-PCA-MLP (Fig. 2.7C-D). Therefore, the standard deviation of the accuracy and the loss in the steady state of the CNN training was significantly larger than those of VGG19-PCA-MLP (p-values: 2.30 × 10-107 for the loss, and 1.03 × 10-255 for the accuracy by two sample F-test) (Fig. 2.7E-F). Since the CNN has much more parameters to learn than VGG19-PCA-MLP, the cost function of the validation set may have much more local minima than that of the training set, which makes the validation loss and accuracy fluctuating during the training process. This suggests that DTL is more robust to the data variability and can produce a more generalizable classifier.

Table 2.10. P-values of the hypothesis testing of the differences in the performance measures between VGG19-PCA-MLP and CNN in NB classification. Wilcoxon rank-sum test was used.

	Accuracy	Sensitivity	Specificity	Kappa	RCI
P-values	0.3571	0.3361	0.8707	0.3507	1

The DTL also used significantly less computational resources than the CNN. For one-time training, the VGG19-PCA-MLP took 30% less time than the CNN (Fig. 2.7G; NVIDIA GTX 1080Ti was used). In the VGG19-PCA-MLP training, the majority of time was spent in the feature extraction (VGG-PCA) rather than MLP training (Fig. 2.7G). Once the features of the training set were extracted, the repeated training was highly efficient whereas the CNN training required the feature extraction in every step of the training. Optimizing VGG19-PCA-MLP was much more efficient compared to the CNN, which could allow for training VGG19-PCA-MLP in a computation limited POC devices. Moreover, combining the automatic cell candidate identification and our DTL based prediction, it took 7.7 seconds to process the whole FOV image (3000 × 3500 pixels, the number of cell candidate: 100). In summary, these results demonstrate the feasibility of hologram classification without reconstruction, simplifying the workflow and decreasing the computational cost for a POC application.

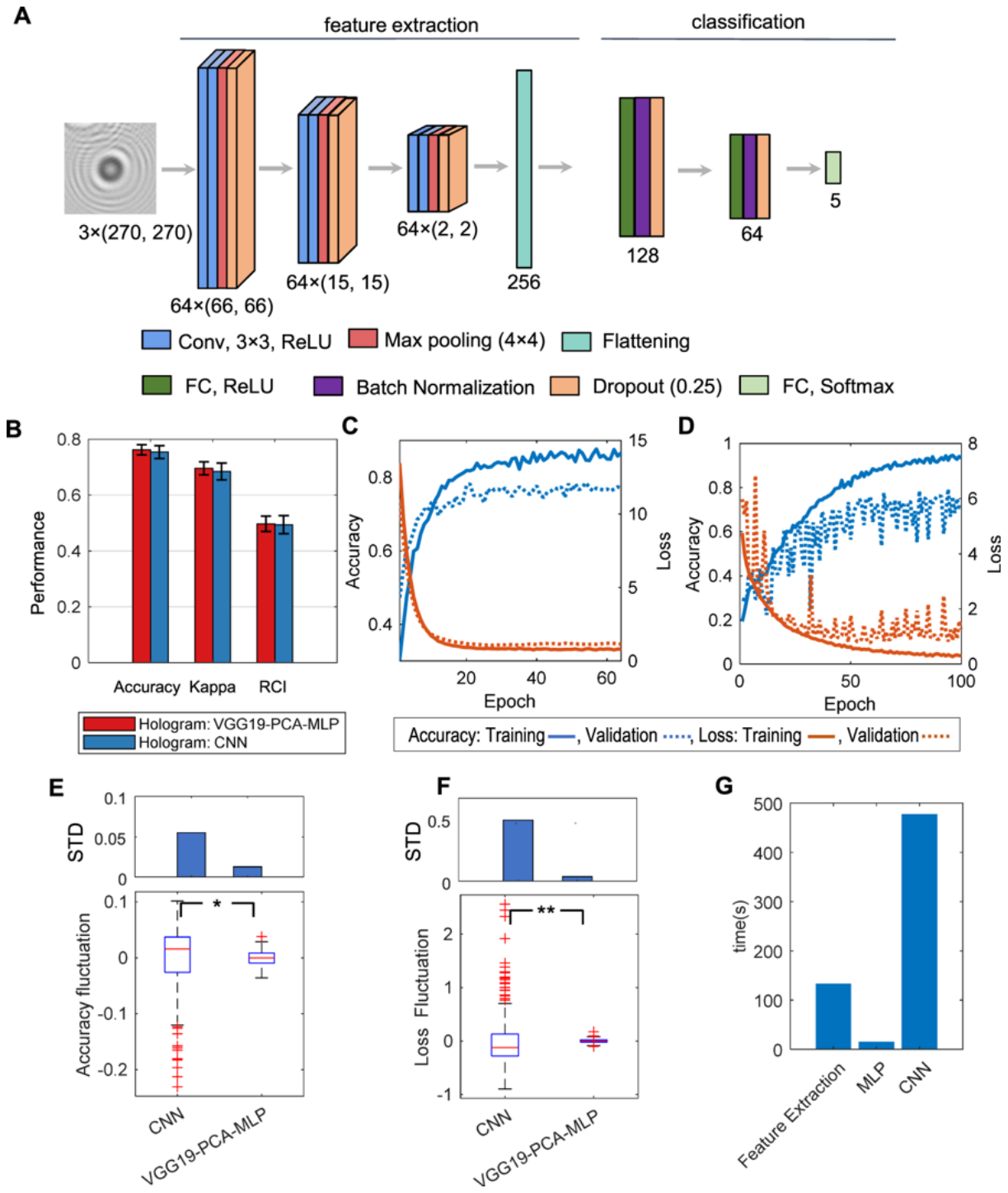


Fig. 2. 7 Classification performance of convolution neural networks (CNN). (A) The structure of the CNN used in this study. (B) Performance comparison between VGG19-PCA-MLP and CNN in the case of NB and holograms. (C, D) Training curves of VGG19-PCA-MLP (C) and CNN (D). (E, F) Fluctuations and standard deviations of the validation accuracy (E) and the validation loss (F) during the training * and ** indicate the statistical significance with p-value, 2.30 × 10−107 and 1.03 × 10−255 respectively by two sample F-tests. (G) Comparison of computing times of the feature extraction (VGG19), MLP training, and CNN training (64 epochs).

2.4 Discussion

We have demonstrated that DTL approaches can effectively classify holograms of bead-bound cells without reconstructing original object images. The conventional reconstruction involves heavy computation, executing iterative phase recovery processes. Our DTL approach requires much less computational power, which could allow for POC devices to train and predict raw holograms.

Intriguingly, our neural networks reliably handled overlapping interference patterns among cells or between cells and unbound beads. In our training set, the target cells were positioned at the centers of the images and other cells or unbound beads were away from the image centers. More than 70% of intensity is concentrated in the first inner circle of a hologram, whereas interferences between two holograms usually happen in the fringes and have much weaker signal strength. Conceivably, the trained networks placed more weight on the hologram center, effectively ignoring fringe patterns.

Our DTL approach could offer appealing new directions to further advance LDIH: (i) deep learning-based training/classification can be executed at the local device level without complex computation; (ii) not relying on high-resolution reconstructed images, the classification network is robust to experimental noises such as reconstruction errors or artifacts; and (iii) the network is elastic and can be continuously updated for higher accuracy in POC devices. With these merits, we envision that the developed ML networks will significantly empower LDIH, realizing a truly POC diagnostic platform.

Literature Cited

1. Garcia-Sucerquia, J. et al. Digital in-line holographic microscopy. Appl Opt 45, 836-850 (2006).
2. Greenbaum, A. et al. Imaging without lenses: achievements and remaining challenges of wide-field on-chip microscopy. Nat Methods 9, 889-895, doi:10.1038/nmeth.2114 (2012).
3. Im, H. *et al.* Digital diffraction analysis enables low-cost molecular diagnostics on a smartphone. *Proc Natl Acad Sci U S A* **112**, 5613-5618, doi:10.1073/pnas.1501815112 (2015).
4. Xu, W., Jericho, M. H., Meinertzhagen, I. A. & Kreuzer, H. J. Digital in-line holography for biological applications. *Proc Natl Acad Sci U S A* **98**, 11301-11305, doi:10.1073/pnas.191361398 (2001).
5. Gurkan, U. A. *et al.* Miniaturized lensless imaging systems for cell and microorganism visualization in point-of-care testing. *Biotechnol J* **6**, 138-149, doi:10.1002/biot.201000427 (2011).
6. Greenbaum, A. *et al.* Increased space-bandwidth product in pixel super-resolved lens free on-chip microscopy. *Sci. Rep.* **3**, 1717, doi:10.1038/srep01717 (2013).
7. Zhu, H., Isikman, S. O., Mudanyali, O., Greenbaum, A. & Ozcan, A. Optical imaging techniques for point-of-care diagnostics. *Lab Chip* **13**, 51-67, doi:10.1039/c2lc40864c (2013).
8. Fienup, J. Phase retrieval algorithms: a comparison. *Appl Opt* **21**, 2758-2769, doi:10.1364/AO.21.002758 (1982).

9. Mudanyali, O., Oztoprak, C., Tseng, D., Erlinger, A. & Ozcan, A. Detection of waterborne parasites using field-portable and cost-effective lensfree microscopy. *Lab Chip* **10**, 2419-2423,
10. Mudanyali, O. *et al.* Compact, light-weight and cost-effective microscope based on lensless incoherent holography for telemedicine applications. *Lab Chip* **10**, 1417-1428,
11. Gerchberg, R. & Saxton, W. A practical algorithm for the determination of phase from image and diffraction plane pictures. *SPIE milestone series MS* **93**, 306-306 (1994).
12. Fienup, J. R. Reconstruction of an object from the modulus of its Fourier transform. *Optics Letters* **3**, 27-29 (1978).
13. Latychevskaia, T. & Fink, H.-W. Solution to the twin image problem in holography. *Physical Review Letters* **98**, 233901 (2007).
14. Rivenson, Y., Zhang, Y., Gunaydin, H., Teng, D. & Ozcan, A. Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light: Science & Applications*, doi:10.1038/lsa.2017.141 (2017).
15. Shen, D., Wu, G. & Suk, H.-I. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering* **19**, 221-248 (2017).
16. Esteva, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115-118, doi:10.1038/nature21056 (2017).
17. Gulshan, V. *et al.* Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* **316**, 2402-2410, doi:10.1001/jama.2016.17216 (2016).
18. Ehteshami Bejnordi, B. *et al.* Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA* **318**, 2199-2210, doi:10.1001/jama.2017.14585 (2017).
19. Pratt, L. Y. Discriminability-based transfer between neural networks. *Advances in Neural Information Processing Systems.* 204-211 (1993).
20. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems.* 3320-3328 (2014).
21. Razavian, A. S., Azizpour, H., Sullivan, J. & Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition, *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on.* 512-519 (IEEE) (2014).
22. Donahue, J. *et al.* Decaf: A deep convolutional activation feature for generic visual recognition, *International Conference on Machine Learning.* 647-655 (2014).
23. Oquab, M., Bottou, L., Laptev, I. & Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on.* 1717-1724 (IEEE) (2014).

24. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. *European Conference on Computer Vision.* 818-833 (Springer) (2014).
25. Choi, J. Y. *et al.* Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database. *PLoS One* **12**, e0187336, doi:10.1371/journal.pone.0187336 (2017).
26. LeCun, Y. *et al.* Handwritten digit recognition with a back-propagation network, *Advances in Neural Information Processing Systems*, 396-404 (1990).
27. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436-444, doi:10.1038/nature14539 (2015).
28. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems.* 1097-1105 (2012).
29. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations* (2015).
30. Deng, J. *et al.* Imagenet: A large-scale hierarchical image database, *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* 248-255 (IEEE) (2009).
31. Fung, J. *et al.* Measuring translational, rotational, and vibrational dynamics in colloids with digital holographic microscopy. *Opt Express* **19**, 8051-8065, doi:10.1364/OE.19.008051 (2011).
32. Cheong, F. C. *et al.* Flow visualization and flow cytometry with holographic video microscopy. *Optics Express* **17**, 13071-13079 (2009).
33. Maaten, L. v. d. & Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research* **9**, 2579-2605 (2008).
34. Cohen, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**, 37-46 (1960).
35. Statnikov, A., Wang, L. & Aliferis, C. F. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics* **9**, 319, doi:10.1186/1471-2105-9-319 (2008).
36. Bhattacharyya, P., Sindhwani, V. & Rakshit, S. Information Theoretic Feature Crediting in Multiclass Support Vector Machines, *Proceedings of the First SIAM International Conference on Data Mining.* 1-18 (2001).

Chapter 3

HACKS: Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging

Cell protrusion is morphodynamically heterogeneous at the subcellular level. However, the mechanism of cell protrusion has been understood based on the ensemble average of actin regulator dynamics. Here, we establish a computational framework called HACKS (deconvolution of heterogeneous activity in coordination of cytoskeleton at the subcellular level) to deconvolve the subcellular heterogeneity of lamellipodial protrusion from live cell imaging. HACKS identifies distinct subcellular protrusion phenotypes based on machine-learning algorithms and reveals their underlying actin regulator dynamics at the leading edge. Using our method, we discover “accelerating protrusion”, which is driven by the temporally ordered coordination of Arp2/3 and VASP activities. We validate our finding by pharmacological perturbations and further identify the fine regulation of Arp2/3 and VASP recruitment associated with accelerating protrusion. Our study suggests HACKS can identify specific subcellular protrusion phenotypes susceptible to pharmacological perturbation and reveal how actin regulator dynamics are changed by the perturbation.

All the listed results have been published in “**Wang C**, Choi H J, Kim S J, et al. Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging[J]. Nature communications, 2018, 9(1): 1688”. As the first author, I initiated the project, designed the algorithm of the time series clustering, performed the correlation analysis, and drug response analysis, and wrote the final version of the manuscript and supplement.

3.1 Introduction

Cell protrusion is driven by spatiotemporally fluctuating actin assembly processes, and is morphodynamically heterogeneous at the subcellular level[1-3]. Elucidating the underlying molecular dynamics associated with subcellular protrusion heterogeneity is crucial to understanding the biology of cellular movement since protrusion determines the directionality and persistence of cell movements or facilitates the exploration of the surrounding environment[4]. Recent studies of the vital roles of cell protrusion in tissue regeneration[5,6], cancer invasiveness and metastasis[7-9], and the environmental exploration of leukocytes[10] further emphasize the physiological and pathophysiological implication of understanding the fine molecular details of protrusion mechanisms. Although there has been considerable progress in analyzing individual functions of actin regulators, the precise understanding of how these actin regulators are spatiotemporally acting in cell protrusion is still limited. Moreover, it is a formidable task to dissect the actin regulator dynamics involved with cell protrusion because such

dynamics are highly heterogeneous and fluctuate on both the micron length scale and the minute time scale[11-13].

Although advances in computational image analysis on live cell movies have allowed us to study the dynamic aspects of molecular and cellular events at the subcellular level, the significant degree of heterogeneity in molecular and subcellular dynamics complicates the extraction of useful information from complex cellular behavior. The current method of characterizing molecular dynamics involves averaging molecular activities at the cellular level, which significantly conceals the fine differential subcellular coordination of dynamics among actin regulators. Over the past decade, hidden variable cellular phenotypes in heterogeneous cell populations have been uncovered by applying machine learning analyses[14,15]; however, these analyses primarily focused on static datasets acquired at the single-cell level, such as immunofluorescence[16], mass cytometry[17], and single-cell RNA-Seq[18] datasets. Although some studies have examined the cellular heterogeneity of the migratory mode[19,20], subcellular protrusion heterogeneity has not yet been addressed. Moreover, elucidating the molecular mechanisms that generate each subcellular phenotype has been experimentally limited because it is a challenging task to manipulate specific subclasses of molecules at the subcellular level with fine spatiotemporal resolution.

To address this challenge, we develop a machine learning-based computational analysis pipeline that we have called HACKS (deconvolution of Heterogeneous Activity in Coordination of cytosKeleton at a Subcellular level) (Fig. 4.1) for live cell imaging data by an unsupervised machine learning approach combined with our local sampling and registration method[13]. HACKS allows us to deconvolve the subcellular heterogeneity of protrusion phenotypes and statistically link them to the dynamics of actin regulators at the leading edge of migrating cells. Based on our method, we quantitatively identify subcellular protrusion phenotypes from highly heterogeneous and non-stationary edge dynamics of migrating epithelial cells. Each protrusion phenotype is demonstrated to be associated with the specific temporal coordination of the actin regulators at the leading edge. Analyzing pharmacologically perturbed cells further verifies that the fine temporal coordination of the actin regulators is required to generate specific subcellular protrusion phenotypes.

3.2 Materials and Methods

3.2.1 Experimental Materials

3.2.1.1 Cell culture and drug treatment The cell culture and live cell imaging procedures were followed according to the previous studies[13]. All imaging was performed in imaging medium (Leibovitz's L-15 without phenol red, Invitrogen) supplemented with 10% FBS, 0.1 mg ml^{-1} streptomycin, 100 U ml^{-1} penicillin, 0.45% glucose, 1.0 U ml^{-1} Oxyrase (Oxyrase Inc.) and 10mM Lactate. Cells were then imaged

at 5 seconds intervals for 1000 s using a 60x, 1.4 NA Plan Apochromat objective for spinning disk confocal microscopy.

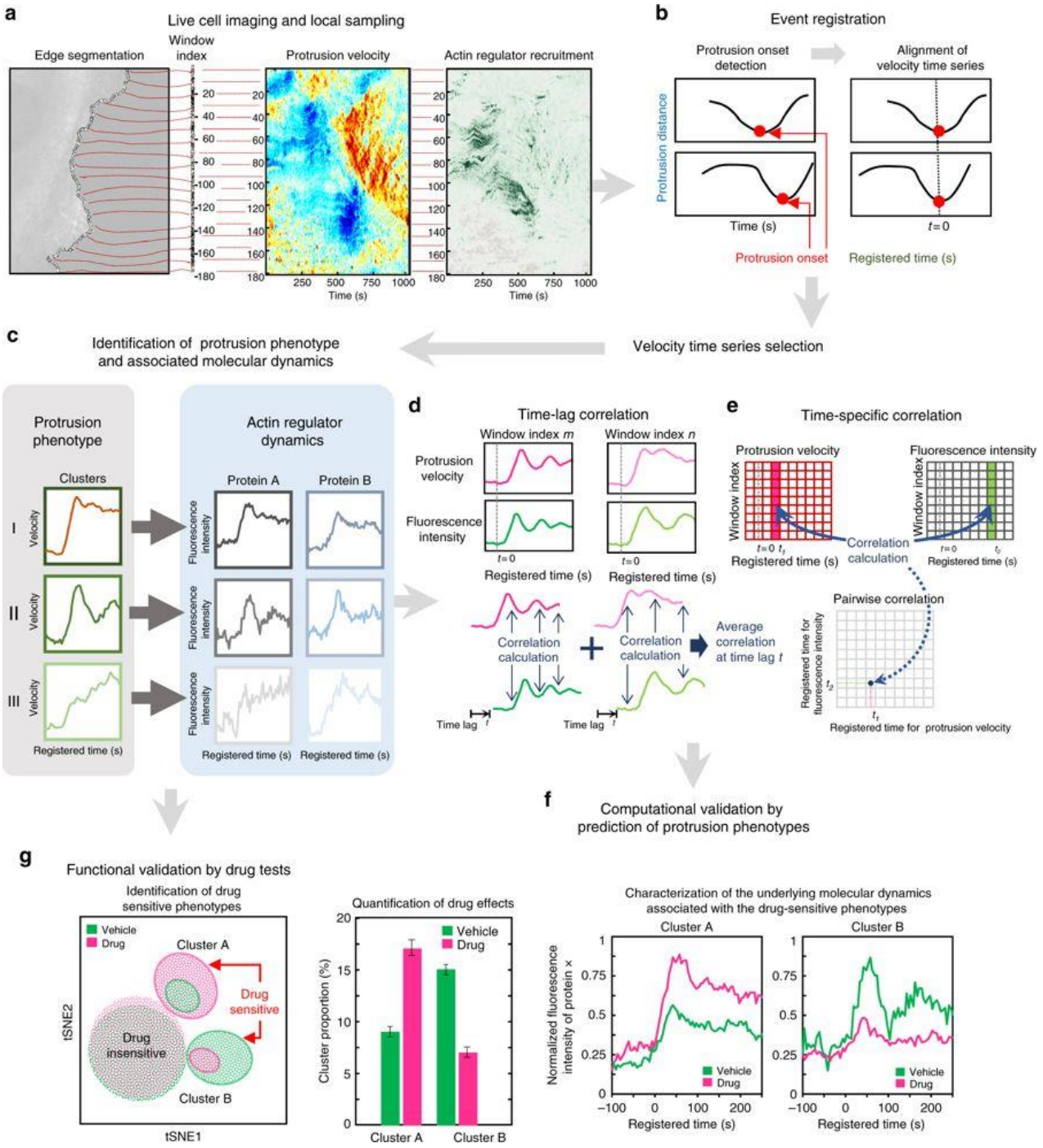


Fig 3.1 Schematic representation of the analytical steps of HACKS.

PtK1 cells were cultured in Ham's F12 medium (Invitrogen) supplemented with 10% FBS, 0.1 mg ml^{-1} streptomycin, and 100 U ml^{-1} penicillin. For the characterization of actin regulator dynamics (Fig. 3.2 and 3), cells were transfected with one of the DNA constructs including HaloTag-VASP (N-term), HaloTag-Arp3 (C-term), SNAP-tag-Actin, empty HaloTag by electroporation using the Neon transfection system (Invitrogen) according to manufacturer's instructions (1 pulse, 1400 V, 20 ms) and were grown on acid-

washed glass #1.5 coverslips for 2 days before imaging. Prior to imaging, expressed HaloTag or SNAP-tag fusion proteins were labeled with HaloTag-TMR ligand (Promega) or SNAP-tag-TMR (New England BioLabs) ligand according to the manufacturer's instructions. PtK1 cells were acquired from Gaudenz Danuser lab. They were routinely tested for mycoplasma contamination.

MCF10A cells were cultured in low - glucose DMEM:Ham's F12 nutrient media supplemented with 5% horse serum, 10 mM HEPES pH 7.4, and a growth factor cocktail including 20 ng ml^{-1} EGF, 10 $\mu g\ ml^{-1}$ insulin, 0.5 $\mu g\ ml^{-1}$ hydrocortisone, and 100 ng ml^{-1} cholera toxin. Cells were grown on 27mm glass bottom dishes (Thermo Scientific, cat. #150682.) for two days. Cells were serum starved for 24hrs and stimulated with growth media containing 10% horse serum before imaging. For plasma membrane staining, cells were incubated with 5$\mu g\ ml^{-1}$ CellMask Orange (Invitrogen) for 5 min followed by manufacturer's protocol. MCF10A cells were acquired from Joan Brugge lab. They were routinely tested for mycoplasma contamination.

For the drug treatment experiments (Fig. 3.5-7), PtK1 cells were grown on 27mm glass bottom dishes (Thermo Scientific, cat. #150682.) for two days and stained with 5$\mu g\ ml^{-1}$ CellMask Deep Red (Invitrogen) following manufacturer's protocol. GFP-Arp3 expressing PtK1 cells were further selected by G418 before imaging. For Arp2/3 inhibition experiments, cells were incubated with 50nM of CK666 or CK689 (EMD Millipore) for an hour before imaging. For Cytochalasin D experiments, cells were incubated with DMSO or Cytochalasin D (Sigma) for half an hour before imaging.

3.2.1.2 Light microscopy All microscopy was performed using the set up as follows: Nikon Ti-E inverted motorized microscope (including motorized focus, objective nosepiece, fluorescence filter turret, and condenser turret) with integrated Perfect Focus System, Nikon Plan Apo 1.4 NA DIC optics (60x), Yokogawa CSU-X1 spinning disk confocal head with manual emission filter wheel with Spectral Applied Research Borealis modification, Spectral Applied Research custom laser merge module (LMM-7) with AOTF and solid state 445nm (200mW), 488nm (200mW), 514nm (150mW), 561nm (200mW), and 637 nm (140mW) lasers, Semrock 405/488/561/647 and 442/514/647 dichroic mirrors, Ludl encoded XY stage, Ludl piezo Z sample holder for high speed optical sectioning, Prior fast transmitted and epi-fluorescence light path shutters, Hamamatsu Flash 4.0 LT sCMOS camera, 37°C microscope incubator enclosure with 5% CO2 delivery (In Vivo), Molecular Devices MetaMorph v7.7, TMC vibration-isolation table.

3.2.1.3 Immunofluorescence PtK1 cells were seeded on cover slips coated with poly-D-lysine. Prior to fixation, cells were incubated with 100nM DMSO or Cytochalasin D for 30 minutes. After drug treatment, cells were fixed with 4% paraformaldehyde in PBS, permeabilized by incubation with 0.1% Triton X-100 in PBS, and subsequently blocked with 1% BSA in PBS for 1 h. To verify the cellular localization of p-VASP and F-actin, cells were incubated with mouse anti-p-VASP antibody (Santa Cruz, sc-365564) and

Alexa Fluor 647 Phalloidin (ThermoFisher, A22287) 1 h in the dark. The cells were washed with PBS for three times and incubated with anti-mouse Alexa Fluor 488 (Invitrogen) for 1 h in the dark. The Ptk1 cells were subsequently washed with PBS for three times and mounted with Gold antifade reagent (Invitrogen). Imaging was performed using the same spinning disk confocal microscope.

3.2.1.4 Plasmid construction Mouse VASP was subcloned into pFN21A vector (Promega) containing an N-terminal fusion to HaloTag. Human Arp3 was subcloned into the pFC14K vector (Promega) containing a C-terminal fusion to HaloTag according to the manufacturer's instructions. A SNAP-tag-actin in C1-vector with a truncated CMV promoter (kindly provided by Martin Schwartz) was used. GFP-Arp3 was a gift from Matthew Welch (Addgene plasmid # 8462).

3.2.2 Details of HACKS

3.2.2.1 Local sampling and event registration. Using a custom-built software package[11,13] written in MATLAB (MathWorks, MA, USA), we performed the following computational procedures. The threshold-based method was used to segment cell edges in the fluorescence images, and the cell edge velocity was calculated by tracking the cell edges using a mechanical model[11]. The software generated probing windows whose initial size was 500 nm by 500 nm along the cell boundary to locally sample the protrusion velocity and fluorescence intensity. The number of probing windows then maintained constant throughout the movie. The local protrusion velocity and fluorescence intensity were quantified by averaging the values within probing windows. By repeating this procedure in each frame of the time-lapse movies, we acquired the time series of protrusion velocities and fluorescence intensities.

We then identified significant protrusion events on a per-window basis. To reduce the effects of random fluctuations in the protrusion velocity time series, we obtained an edge displacement time series for a particular window by integrating the protrusion velocity over time. The noise of the time series was removed with a smoothing spline filter using the Matlab function *csaps()* and a smoothing parameter of 0.01. The small protrusion and retraction events considered insignificant in terms of the overall cell edge movement were then further eliminated as follows. First, we identified local maxima/minima (protrusion/retraction onsets) at the edge displacement time series using the Matlab function *findpeaks()* and calculated the net protrusion/retraction distances for each event. A previous study using the same PtK1 cells showed that the distribution of distances could be decomposed into two exponential distributions, indicating small fluctuations and large movement during protrusion and retraction events[13]. Thus, small events whose protrusion distances were less than 720 nm (10 pixels in length) were discarded from the analysis. In addition, we eliminated short-term switches between the protrusion and retraction phases within 50 seconds. After these insignificant events were removed1, the remaining protrusion onsets were used for event registration.

The protrusion velocity and fluorescence intensities over time in individual windows were registered by aligning the protrusion onset at t = 0. After the registration, the negative time indicates the retraction phase, and the positive time indicates the protrusion phase. Time series in negative time were limited by the preceding protrusion onset, and time series in positive time were limited by the subsequent retraction onset.

3.2.2.2 Interpolating missing values. Because of image noise, the software can produce abnormal data in a rare case. In this case, values can be missed from a time series, and the following strategy was applied to treat these missing values: For each edge velocity or fluorescence intensity sample, the entire time series was discarded if the length of continuous missing values is longer than a threshold (here, we used the value 8). Otherwise, the average of four values before and after the missing value was used to individually estimate the value for this location.

3.2.2.3 De-noising the samples by Empirical Mode Decomposition. For each registered sample, the edge displacement was calculated from the edge velocity using the Matlab function *trapz()*. Empirical Mode Decomposition (EMD)[21] was then applied to the transformed protrusion edge displacement to remove noise. Finally, the denoised velocity was calculated from the denoised displacement using the Matlab function *diff()*.

Cell movement is highly non-stationary, and Empirical Mode Decomposition (EMD)[21] is a local and data-driven de-noising method to decompose non-stationary signals into a series of intrinsic components. The general procedure of EMD can be described as follows:

1). Identify all the extremes (minima and maxima) of sample *d(t)*;

2). Connect the local maximum points and local minimum points separately using an interpolation method to generate the envelope, *e(t)*;

3). Compute the average of envelopes, $avg(t) = (e_{min}(t) + e_{max}(t))/2$;

4). Eliminate the average signal of the envelope from the sample *d(t)* to obtain the residual: $avg(t) = (e_{min}(t) + e_{max}(t))/2$;

5). Iterate from steps 1) to 4) on the residual *m(t)* until the *avg(t)* equals zero.

After EMD, the original signals can be decomposed into intrinsic mode functions (IMF) without any loss of information, and the residue is called the trend. For each component, a de-trended fluctuation analysis was used to measure the self-affinity as the fractal scaling index (α), which estimates the fractal-like autocorrelation properties. The value of α inversely correlated with the possibility that the component is originated from noise. In our procedure, the code was obtained from a previous publication[50], and the value of α was empirically set to 0.33 to balance the maintaining information and trimming noise.

3.2.2.4 Determining the time interval for the clustering analysis. The duration of cell protrusion is heterogeneous, and some protrusion events were not completely recorded because of the finite length of the movies. Our clustering analysis focuses on the equal length of time series data. Therefore, time series shorter than a certain threshold were discarded from the analysis. In order to determine the suitable threshold for our specific data, the best scenario is that the threshold should be selected as the optimized solution of maximizing the multiplication of these two factors: the number of samples and the left temporal length of each sample. After optimizing these two factors with equal weight, the best threshold is approximately 50. After this step, more than 60% of samples remain for further analysis. The exact threshold, which is near 50, is finally decided based on convenience for further analysis. Considering the ambiguity of detected protrusion onsets, five frames before the protrusion onset were also included for the further analysis. Therefore, the time series for the analysis consisted of 56 frames, which included the previous 5 frames before the protrusion onset and 51 frames after the protrusion onset.

3.2.2.5 Representing the velocity by Symbolic Aggregate approximation (SAX). The dimensionality of times series samples consisting of 56 frames remained still high, and we were interested in general patterns over large time scales. Therefore, the dimensionality of time series data should be reduced by some dimension reduction methods. To this end, we applied SAX (Symbolic Aggregate approXimation)[51] to our time series dataset to reduce dimensionality and discretize the data. The general procedure of SAX is summarized as follows:

1). Manually determine the reduced dimension, N, and the symbolic number, M (the number of discretization levels).

2). For each normalized sample, the time series data over the entire time range are pooled together and fitted to a Gaussian distribution. The entire time series is divided into M ranges with equal probabilities of the fitted Gaussian distribution. We represent the values of each range a symbol accordingly.

3). Subsequently, the time series is divided into N intervals along over time. The average value is calculated in each interval to represent the raw time series data.

4). Finally, the average value in each interval is represented by the symbol defined in step 2.

5). Iterate from step 2) and 4) until all samples are represented.

After the SAX representation, all time series data are reduced to low-dimensional (N) symbolic series data, which are used in further analyses. In the current experiment, M and N were both empirically set to four. Here, 4 symbols that range from 0 to 3 will be used to calculate the autocorrelation coefficients. In addition, as an implicit benefit, the symbol representation process in SAX also removes local noise.

3.2.2.6 Calculating the sample dissimilarity. To measure the dissimilarity of two time series, the original description of SAX representation proposed an approximate Euclidean distance of SAX as a dissimilarity measure[22]. Instead, we used the dissimilarity measure based on estimated autocorrelation functions (ACF) [52]. First, the estimated autocorrelation vector was calculated, and the square Euclidean distance between the autocorrelation coefficients was then used to measure the dissimilarity of two samples as follows:

$$d_{\mathrm{ACF}}(X,Y) = \sum_{i=1}^{L} (X_i - Y_i)^2$$

In our implementation, the ACF distance was calculated using the TSdist R package[53]. Besides, in order to evaluate the performance of ACF measurement, we compared the clustering performance using different dissimilarity measures in Supplementary Fig. 3.1.

3.2.2.7 Clustering the velocities by Density Peak. After we calculated the pairwise dissimilarity of the time series, we performed a clustering analysis using the Density Peak clustering algorithm[23]. It is desirable that cluster centers have local density maxima and are separated from other dense regions in feature space. Based on this idea, Density Peak can generate a density-distance map that can be used to determine the number of clusters and cluster centers. In addition, Density Peak can build up hierarchical tree structures of clusters by linking the samples with higher density but lower distance. Based on the selected number of clusters and cluster centers, the samples in the hierarchical tree can be divided into several clusters.

The procedure to generate the density-distance map is as follows: Each sample is represented by two parameters: local density and minimum distance measurement. The local density of each sample is estimated by the crowdedness of samples in its neighboring region. The minimum distance measurement is the distance of the closest samples with higher density. By plotting these two parameters in two dimensions, we build up the density-distance map. Based on the definition, the samples with a high density and distant from other samples with higher density are localized in the upper-right region of the density-distance map. Therefore, the sparse samples in the upper-right region will be selected as cluster centers visually, and the number of these cluster centers are determined as the number of clusters. Finally, the hierarchical tree is divided into several disconnected sub-trees as clusters.

In our implementation, the density around each sample was determined by calculating the sum of distances with the Gaussian Kernel of the manually selected radius as follows:

$$\rho(S_i) = \sum_{k=1, i\neq k}^{N} e^{(-\frac{d(S_i,S_k)}{dc})^2}$$

Here, *dc*, represented by the ratio of the range of distance matrix, is selected by a grid search method of the ratio to get the good performance of density-distance map. The number of clusters is manually selected by the visual inspection of the density-distance map, which is the same as the idea of the original paper. Besides, in order to further confirm the number of clusters suggested by density-distance map, we also applied another three criteria: Davies-Bouldin Index (DBI), Average Silhouette and Calinski-Harabasz pseudo F-statistic to evaluate the number of clusters implemented in ClusterSim Package[54].

3.2.2.8 Validating clustering results. We used the following methods to validate our clustering results.

- Ordered Dissimilarity Map: After clustering, the distances between samples within the same clusters should be smaller than those between samples in different clusters. Therefore, after the samples are grouped by cluster indices and ordered by distance, the distance map can be visualized as blocks along the diagonal. In addition, because the input of the Density Peak clustering method is only a distance matrix, the ordered dissimilarity map will be suitable to evaluate clustering results to show several blocks along the diagonal.

- MDS: Classical multidimensional scaling (MDS)[55] is a method to visualize the similarity of individual samples in a dataset based on the distance dissimilarity matrix. The MDS algorithm aims to place each sample in a lower dimensional space under the constraint that the between-sample distances are preserved as much as possible. Here, the Classical Multidimensional Scaling method in Matlab function, *cmdscale()*, was used.

- t-SNE: t-SNE (t-distribution Stochastic Neighboring Embedding)[56] is an advanced dimensionality reduction technique and a particularly suitable visualization method for high-dimensional datasets. In t-SNE, the similarity of two samples is the conditional probability density to measure the neighborhood under the t-distribution centered at each sample. The ultimate goal is to minimize the total mismatch between the conditional probability of piecewise samples under the t-distribution determined by calculating the sum of Kullback-Leibler (K-L) divergences over all samples. Here, the parameters (final dimension, initial-dimension, perplexity) of t-SNE were (2, 10, 20), indicating that the dataset was first reduced to 10 dimensions and then mapped to two dimensions by optimizing the K-L divergences.

- Silhouette Plot: Silhouette plots[57] are used to validate the consistency within clustered data. For each sample, *a(i)* represents the average dissimilarity within the same cluster, whereas *b(i)* represents the lowest average dissimilarity within any other cluster. Finally, the silhouette value of the sample, *i*, is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}$$

The range of *s(i)* is [-1,1], and larger values indicate a better clustering performance.

3.2.2.9 Normalizing Actin, Arp2/3, and VASP fluorescence signals. Because differences in the expression levels of fluorescent proteins and their endogenous non-fluorescent proteins are not known, we cannot average the registered time series of raw fluorescence intensity. Moreover, we aimed to determine the recruitment pattern for a fluorescent protein independent of the absolute level. Therefore, before a protrusion event was registered, we separately normalized the intensity time series of each window by min-max scaling as follows:

$$I_{norm}(w,t) = \frac{I(w,t) - \min\left(I(w,t)\right)}{\max(I(w,t)) - \min(\mathrm{I}(w,t))} * 1000$$

3.2.2.10 Correlation analysis. The time lag correlation analysis between two time series as a function of time lag is used to exploit the temporal relationship[12]. Pearson's correlation coefficients were calculated using the time series of only protrusion segments (after protrusion onsets). The 95% confidence intervals for the average correlation were calculated by bootstrap resampling (Matlab function *bootci()*).

The time-specific correlation analysis between two activities (velocity and regulator) is used to exploit the spatiotemporal variation [13]. After the protrusion velocity and fluorescence intensities were registered with respect to protrusion onset at $t = 0$, Pearson's correlation coefficients (Matlab function *corrcoef()*) between the fluorescence intensity at t_1 and protrusion velocity at t_2 across the samples were calculated across the time points, where t_1 and t_2 were measured relative to the protrusion onset. Two sample K-S (Kolmogorov-Smirnov) test was used to test the significance of the maximum correlation value in time specific correlation and Benjamini-Hochberg procedure for controlling the false discovery rate (FDR) was used to show the significance of the time-specific correlation.

3.2.2.11 Clustering analysis of drug-treatment data. When we compared the proportions of each cluster with and without drug treatment, we pooled the registered protrusion velocity time series from the control and drug-treated experiments to maintain the same cluster boundaries, and then applied our time series clustering to them under the same clustering criteria. For DMSO/CyD treatment, the parameter of Density Peak cluster is 0.71. For CK689/CK666 treatment, the parameter of Density Peak cluster is 0.46.

For the GFP-Arp2/3 experiments, where we compare the temporal pattern in the similar clusters, we applied our time series clustering to the control and CyD treatment experiment individually. The parameters of Density Peak cluster are both 0.71.

3.2.2.12 Identification of the drug-sensitive phenotypes. We pooled the control and drug-treatment data and visualized the data distribution of denoised velocity time series using t-SNE[56]. The initial dimension and the perplexity of the t-SNE were 30 and 50. Using the t-SNE plot, we visually identify the

drug-susceptible regions where the data from the drug treatment were depleted in comparison to the control. By overlaying the cluster assignments, then we identify which clusters were mainly affected by the drug treatment. We extracted the data belonging to these drug-susceptible clusters and applied community detection method[40] to the selected data to identify the sub-clusters. Then we merged these sub-clusters into two clusters based on the magnitude of the average velocity. In addition, the boundaries of the drug-susceptible regions were considered to finalize the sub-cluster structure. Finally, the cluster proportions with and without drug treatment were compared to validate the drug-sensitive phenotype.

3.2.2.13 Statistical testing of the proportions of drug-sensitive phenotypes. We quantified the drug effect based on the cluster proportion. We counted the number of each cluster in each cell for the control and drug treatment experiments. These numbers were resampled using *bootstrp()* in MATLAB to build 10000 different bootstrapped dataset, and the distribution of the proportions of each cluster in each experimental set was created. Using these distributions, p-values were calculated by estimating the probability that the cluster proportion of one experiment is greater or less than that of the other experiment (one-tailed test). The confidence intervals of each experiment were estimated by the Matlab *bootci()* function.

3.2.2.14 Spatial distribution of subcellular protrusion clusters. We calculated the self-conditional probability that the samples within the same cluster co-exist over the distance as follows. In each iteration, we randomly selected eight movies from the total 36 movies and then sampled 40 frames in each movie. For a certain distance or window gaps, *k*, we calculated the 5-by-5 occurrence matrices, $M_k(\, cl_i, cl_j)$ for different pairs, cl_i, cl_j of five clusters without considering the direction along cell edges. Based on the occurrence matrix, we calculated the conditional probability of each cluster pair for different window gaps as follows.

$$p_k(cl_i \mid cl_j) = \frac{M_k(\, cl_i, cl_j)}{\sum_{cl_i} M_k(cl_i, cl_j)}, \qquad cl_i, cl_j = 1, \dots, 5$$

We averaged the conditional probability $p_k(cl_i \mid cl_j)$with 500 iterations and the confidence intervals of the mean was estimated by bootstrapping (*bootci()* in Matlab).

3.2.2.15 Evaluating different time series clustering methods. To show the effectiveness of our time series clustering, three main components, SAX for dimensional reduction, ACF for dissimilarity measure and Density Peak for clustering, were evaluated by replacing them with different methods as follows.

1) Evaluating the role of SAX: Without using SAX, ACF was directly applied to the denoised velocity dataset to calculate the ACF distances, implemented in the TSdist R package[53]. The Density Peak method was then used for clustering with the cut-off distance, 0.61. Community detection was used for clustering with the number of neighbors, 80.

2) Evaluating the role of ACF distance: The dissimilarity measure was changed from the ACF distance to the distance metric proposed by SAX, which is the lower bound of the true Euclidean distance[51]. Here, 8 was empirically selected as the number of symbols for SAX, and 8 symbols ranging from 0 to 7 were used to calculate the distance. The cut-off distance of the Density Peak clustering was 0.46.

3) Evaluating the role of the combination of SAX and ACF: Without dimensional reduction by SAX, the denoised velocity dataset was directly used to calculate the dissimilarity using the Euclidean distance from the TSdist R package. The Density Peak method was then used for clustering, and the cut-off distance for the Density Peak method was 0.46.

4) Evaluating the role of Density Peak clustering: Instead of Density Peak clustering, a conventional clustering method, K-means, and community detection were used for comparison while all other steps remained unchanged. Since the number of clusters for K-means should be determined first. Here two internal criteria DBI (Davies–Bouldin Index)[24] and Silhouette criteria[57] were used to identify the number of clusters. The number of clusters for K-means was set to the optimal number K=7. Community detection was also applied for comparison using the number of neighbors, 300 or 350 to generate 6 and 5 clusters.

3.2.2.16 Classification analyses of actin regulator intensities. To further investigate the role of VASP in accelerating cell protrusions, we applied the classification approach to the fluorescence intensity time series with their corresponding protrusion clusters. For this purpose, we focused on the classification between the non-accelerating protrusion class (Clusters I/II) and accelerating protrusion class (Cluster III). First, the fluorescence intensity time series was normalized to have a mean of zero and a standard deviation of one for each cell and then normalized again in the same manner for each window. We used three different classification algorithms to measure the performance of the classification, including random forest (RF)[58], support vector machine (SVM)[59], and deep neural networks (DNN)[60]. The inputs of the classifiers were the normalized fluorescence intensities of selected frame intervals based on protrusion onset previously, and the output was the corresponding protrusion class (non-accelerating vs accelerating protrusion). The supervised learning was performed using the Python Scikit-Learn toolkit for RF and SVM[61] and Keras with Theano engines in Python for DNN. Because the number of time series in the non-accelerating protrusion class (Clusters I/II) was larger than those of the time series in the accelerating protrusion class (Cluster III), we under-sampled the accelerating protrusion class so that the number of data points in two classes had the same. For reproducible results, random under-sampling was applied ten times to the non-accelerating protrusion class using the Imbalanced-learn package[62]. Cross-validation was performed with 67% and 33% splitting of each sample dataset for training and testing. Moreover, the cross-validations for each sample were repeated ten times after randomly shuffling the data in each iteration. Hence, we performed the training procedures for each fluorescence intensity

dataset for 100 times. To assess the performance of the classification, we used accuracy (*Nc/N*), where *N* is the number of total sequences and *Nc* is the number of matched sequences between the original and prediction, and Matthews correlation coefficient (MCC) defined as

$$\frac{N_{tp}N_{tn} - N_{fp}N_{fn}}{\sqrt{(N_{tp} + N_{fp})(N_{tp} + N_{fn})(N_{tn} + N_{fp})(N_{tn} + N_{fn})}}$$

Here, N_{tp}, N_{tn}, N_{fp}, *and* N_{fn} are the numbers of true positives, true negatives, false positives and false negatives, respectively. The accuracy and MCC were calculated using the Python Scikit-Learn toolkit, where the parameters used in three classifiers are shown in Supplementary Table 5, which were determined by a grid search approach.

3.2.2.17 Normalizing GFP-Arp2/3 fluorescence signals. In order to quantitatively compare the fluorescence intensity of GFP-Arp2/3 between DMSO and CyD-treated cells, we normalized the Arp3 intensity time series in each cell as follows to make sure that the normalized intensities of protrusion onset in these two cases are similar. In each cell, we manually selected the lamella regions, which do not contain bright fluorescence spots. Then, we calculated the average fluorescent intensity, I_{la} in these regions. We also selected background region outside the cell and the average background intensity, I_b was calculated. Finally, we calculated the GFP-Arp2/3 fluorescent normalized intensity value, I_{norm} from the raw intensity, *I* in each cell as $I_{norm} = (I - I_b)/(I_{la} - I_b)$.

3.2.2.18 General Statistical Methods.

- The sample size was determined as follows. We generally use more than 100 probing windows from multiple cells (see individual figures or figure legends). The number of the probing windows is determined to be sufficient when the averaged time series displays a distinct pattern with variations that substantially exceed the 95% confidence interval.

- Inclusion/exclusion of samples was determined as follows. We visually examined cellular morphology, the level of protein expression, and the number of nuclei in each cell movie. We perform our analysis using the cells with a flat, minimally ruffling morphology and wide leading edges, low expression level of fluorescent proteins, and single nucleus. At this stage, we do not know the cluster distribution along the cell edges and how the protein dynamics would behave. Thus, this data selection can be assumed unbiased for the presented analyses.

- Justification of statistical tests: we used two-sample K-S (Kolmogorov-Smirnov) test implemented with the Matlab function *kstest2()* for statistical tests. The K-S test does not assume the distribution of data. The variances of the data between groups are similar (See each figure). For the multiple hypothesis testing in time-specific correlation analysis, Benjamini-Hochberg procedure for controlling the false discovery rate (FDR) was used to provide stronger control of

the family-wise error rate. The 95% confidence interval of the velocity and normalized intensity was calculated using the bootstrap Matlab function *bootci()*, and the number of bootstrap samples was set to 1000.

3.3 Results

3.3.1 HACKS: Deconvolution of subcellular protrusion heterogeneity

To deconvolve the heterogeneity of the subcellular protrusion activity and their regulatory proteins at fine spatiotemporal resolution, we developed a computational analysis pipeline, HACKS (Fig. 3.1), which is based on an unsupervised machine learning method. HACKS allowed us to (1) identify distinct subcellular protrusion phenotypes based on a time series clustering analysis of heterogeneous subcellular protrusion velocities extracted from live cell movies (Fig. 3.1a-c), (2) associate each protrusion phenotype with pertinent actin regulator dynamics by comparing the average temporal patterns of protrusion velocities with those of actin regulators (Fig. 3.1c), (3) perform highly specified correlation and classification analyses of actin regulator dynamics of protrusion phenotypes to establish their association with fine mechanistic details (Fig. 3.1d-f), and (4) identify specific protrusion phenotypes susceptible to molecular perturbations, and functionally confirm the association between protrusion phenotype and the actin regulator dynamics (Fig. 3.1g). The framework can provide mechanistic insight into how the differential coordination of actin regulator dynamics organizes various subcellular protrusion phenotypes.

3.3.2 A time series clustering analysis of protrusion velocities

Sample videos for the analysis were prepared by taking time-lapse movies of PtK1 epithelial cells expressing fluorescently tagged actin, Arp3, VASP and a cytoplasmic marker, HaloTag, with a spinning disk confocal microscope for approximately 200 frames at 5 seconds per frame[11] (Fig. 3.1a). Each time-lapse movie contains a single cell whose leading edge undergoes protrusion-retraction cycles. After segmenting the leading edge of each cell by multiple probing windows with a size of 500 by 500 nm[13] (Fig. 3.1a, left), time series of velocities[11] and fluorescence intensities of the tagged molecules[12,13] acquired from each probing window were quantified (Fig. 3.1a, center and right). After registering protrusion onset at time zero (t=0), the time-series were aligned using the protrusion onset as a temporal fiduciary[13] (Fig. 3.1b). To ensure a uniform time length of the data for the subsequent clustering analysis, we selected the first 50 frames (250 seconds) of protrusion segments, which is about the average protrusion duration[13] from the pooled velocity time series.

The selected time series of the registered protrusion velocity contained a substantial amount of intrinsic fluctuations, hindering the identification of distinct clusters of similar protrusion activities. Therefore, we first denoised the time series velocity profile using Empirical Mode Decomposition[21] and discretized the data using SAX (Symbolic Aggregate approximation)[22] to reduce the dimensionality and complexity of the data (Supplementary Note 1). We then extracted distinct patterns from fluctuating velocity time series

by combining the autocorrelation distance measure with the Density Peak clustering[23]. The distance measures between different time series were calculated using the squared Euclidean distances between the corresponding autocorrelation functions (ACF) of each discretized time series. This autocorrelation distance partitioned the fluctuating time series of similar underlying patterns into the same clusters, enabling us to identify clusters with clear dynamic patterns (Supplementary Note 1). Following the ACF distance measure, we applied the Density Peak clustering algorithm, which has been shown to be superior to conventional K-means in partitioning data with complex cluster shapes[23]. As a result, the density-distance graph in Fig. 3.2g, where cluster centers are localized in the upper-right region (see Methods for detail), revealed five distinct clusters of subcellular protrusion activities. Using the clustering criteria, Davies-Bouldin Index[24], average silhouette value, and Calinski-Harabasz pseudo F-statistic[25], we also confirmed that the optimal number of clusters was five (Supplementary Fig. 3.3a-c). After the clustering analysis, average protrusion velocities and the 95% confidence intervals of the mean were calculated (Fig. 3.2e). Of note, after we tested different sets of algorithms, we found that ACF distance was the most important factor which allowed us to extract these distinct temporal patterns (Supplementary Fig. 3.1 and 2, Supplementary Note 1). Furthermore, we could not identify substantial differences among the velocity cluster profiles (dotted lines in Fig. 3.3b-e) in each molecule (actin, Arp3, VASP, HaloTag), confirming that our clustering results are not skewed by a specific data set. The numbers of cells and probed windows used in the time series clustering analysis are presented in Supplementary Table 1.

3.3.3 Identification of distinct subcellular protrusion phenotypes

The visual inspection of the average velocity profiles of the identified clusters (Fig. 3.2e) demonstrated that the overall differences among the protrusion phenotypes originated from differences in the timing and number of peaks the velocity reached. Whereas Cluster I did not exhibit dramatic changes in protrusion velocities after reaching its peak at the earlier part of the protrusion segment (Fig. 3.2e), the remaining clusters exhibited substantial acceleration or deceleration in the protrusion velocities with varying timing and number. Clusters II-1, II-2, and II-3 (Fig. 3.2e) exhibited different periodic changes in the acceleration and deceleration of protrusion. Conversely, Cluster III (Fig. 3.2e) demonstrated persistently accelerating behavior where protrusion velocities continued to increase until the late phase of the protrusion. Clusters I, II-1, II-2, II-3 and III comprised 27.7%, 13.3%, 22.7%, 24.5%, and 11.8% of the entire sample, respectively, and individual cells expressing different fluorescent proteins exhibited similar tendencies (Fig. 3.2f, Supplementary Fig. 3.3h), suggesting the intracellular origin of protrusion heterogeneity. Nevertheless, cell-to-cell variability in cluster distribution persisted, suggesting that the clusters may also reflect individual cellular responses to differential cellular contexts or microenvironments.

The validity of our clustering result was confirmed by visually inspecting the velocity activity map (Fig. 3.2d). Clusters II-1/2/3 (Fig. 3.2d) and III (Fig. 3.2d) exhibited clearly distinguishable patterns, whereas Cluster I (Fig. 3.2d) contained fluctuating velocity profiles (See Supplementary Fig. 3.3g for the full maps). The t-SNE (Fig. 3.2h), multidimensional scaling, silhouette, and order distance plots (Supplementary Fig. 3.3d-f) of the clustering results further confirmed the stability and tightness of Clusters II-1/2/3 and III but suggested residual heterogeneity in Cluster I, which is in agreement with the velocity activity maps (Fig. 3.2d). To quantify the spatial structure of the protrusion phenotypic clusters, we estimated the conditional probability that the same cluster exists over the distance from a given cluster (Fig. 3.2i). As the distance increases between two neighboring clusters, this conditional probability in all clusters decreases to their basal levels of the cluster proportions (Fig. 3.2i). The conditional probability in Cluster II-1/2/3 quickly decreased within 2 mm distance whereas those in Cluster I and III persisted up to 5 mm (Fig. 3.2i). This data suggests that Clusters I and III aggregate and act collectively more so compared to Cluster II-1/2/3. In addition to PtK1 cells, we further performed the same analysis on MCF10A, human mammary epithelial cells. MCF10A also had very similar subcellular protrusion patterns (Supplementary Fig. 3.4), suggesting that these results by HACKS are not limited to a specific cell line.

Notably, this procedure revealed the differential subcellular protrusion phenotypes with distinct velocity profiles using our time series clustering framework. The visualization of the edge evolution (Fig. 3.2a), the cluster assignments evolution (Fig. 3.2b), and the protrusion velocity map (Fig. 3.2c) of the exemplified live cell movie representatively manifested the morphodynamic features of each subcellular protrusion phenotype. Cluster I was named ‘fluctuating protrusion’ because of the irregularity of its velocity profiles. Since Cluster II-1/2/3 clearly exhibited periodic edge evolution, we refer to Cluster II-1/2/3 collectively as ‘periodic protrusion.’ Notably, Cluster III showed accelerated edge evolution, and, therefore, we refer to Cluster III as ‘accelerating protrusion.’

3.3.4 Differential molecular dynamics of actin regulators

We hypothesized that the distinctive subcellular protrusion phenotypes arise from the differential spatiotemporal regulation of actin regulators. Therefore, we next investigated the relationship between the velocity profiles of each protrusion phenotype and the fluctuation of the signal intensities of actin and several actin regulators for each protrusion phenotype. We selected a set of fluorescently tagged molecules to be expressed and monitored; SNAP-tag-actin, HaloTag-Arp3 (tagged on the C-terminus), which represented the Arp2/3 complex involved in actin nucleation, and HaloTag-VASP or GFP-VASP, which represented actin elongation. A diffuse fluorescent marker, HaloTag labeled with tetramethylrhodamine (TMR) ligands[26], was used as a control signal. The fluorescence intensities of each tagged molecule were acquired from each probing window along with the protrusion velocities (Fig. 3.1a). The time-series of the fluorescence intensities of each molecule were then grouped and averaged according to the assigned protrusion phenotype (Fig. 3.1c and Fig. 3.3b-e).

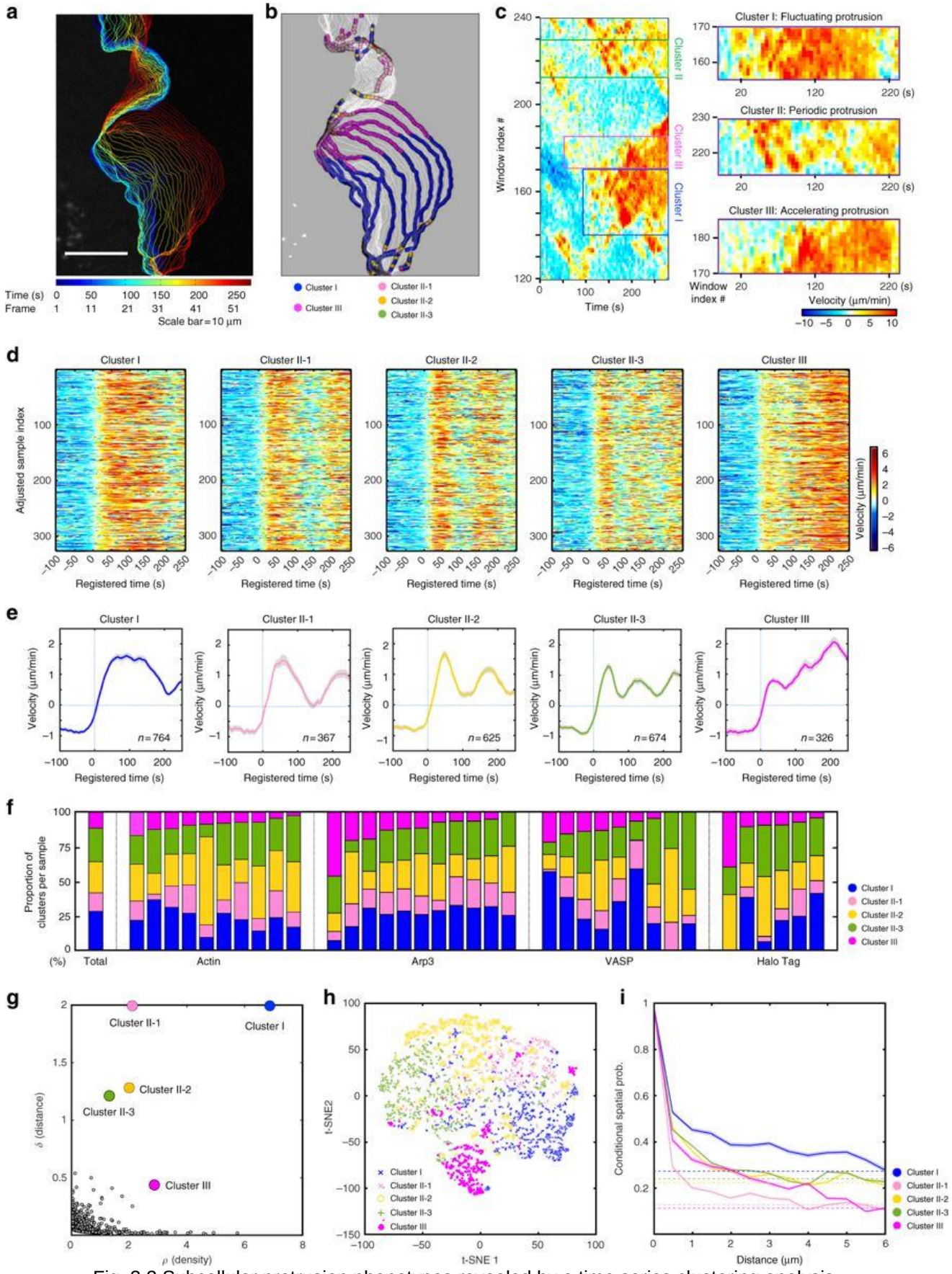


Fig. 3.2 Subcellular protrusion phenotypes revealed by a time series clustering analysis.

Whereas the molecular dynamics of actin, Arp3 and VASP all exhibited patterns similar to those of the velocity profiles in Clusters I and II-1 (Fig. 3.3b-d, Cluster I/II-1 each), the Arp3 temporal patterns became less correlated with those of protrusion velocity in Cluster II-2 and II-3 as the frequency of the oscillation increased (Fig. 3.3c, Cluster II-1/2/3). This demonstrates that underlying molecular temporal patterns can be highly variable depending on dynamic properties of protrusion activities. Intriguingly, Cluster III also exhibited distinctive differential molecular dynamics among the molecules and in relation to velocity profiles (Fig. 3.3a-d, Cluster III each). Specifically, whereas the protrusion velocity continued to increase until the late stages of the protrusion segment in the accelerating protrusion (Fig. 3.3a, Cluster III), the actin fluorescence intensity soon reached its maximum in the early phase and remained constant (Fig. 3.3b, Cluster III). This pattern indicates that edge movement during accelerating protrusion is mediated by the elongation of existing actin filaments rather than *de novo* actin nucleation. Conversely, Clusters I and II-1/2/3 exhibited increased actin intensity at the leading edge along with increased protrusion velocity (Fig. 3.3b, Cluster I, II-1/2/3), indicating that actin nucleation mediates subcellular protrusion.

In accordance with the plateaued actin intensities in Cluster III (Fig. 3.3b, Cluster III), the Arp3 intensity remained constant after reaching its peak in the early protrusion phase (Fig. 3.3c, Cluster III), whereas the VASP intensities began to increase at protrusion onset and continued to increase (Fig. 3.3d, Cluster III). These findings suggest that actin elongation by VASP plays a crucial role in driving accelerating protrusion. Whereas the Arp2/3 complex has been considered as a major actin nucleator that drives lamellipodial protrusion[27], Arp2/3 seemed to play a role in the earlier part of the protrusion in accelerating protrusion. Approximately 50 seconds after protrusion onset, the Arp3 intensity reached its peak (Fig. 3.3c, Cluster III), and the acceleration temporarily stopped (Fig. 3.3a, Cluster III). Notably, the Arp3 intensities began to increase 50 seconds prior to the protrusion onset in Cluster III (Fig. 3.3c, Cluster III), whereas they began to increase at the onset of the protrusion in Clusters I and II-1/2/3 (Fig. 3.3c, Cluster I, II-1/2/3). These findings imply that there exists specific temporal coordination where the Arp2/3 complex nucleates actin networks in the early phase, and VASP then elongates actin filaments to drive the later stages of accelerating protrusion. The specificity of the relationship between the protrusion phenotypes and the underlying molecular dynamics was further validated with a control experiment using HaloTag-TMR (Fig. 3.3e). Diffused cytoplasmic fluorescence did not exhibit any cluster-specific pattern. Instead, it inversely correlated with the protrusion velocity, suggesting that the cell edges become thinner as the protrusion velocity increases[13]. Notably, the differential dynamics of Arp3 and VASP were not observed when the entire time series dataset was ensemble averaged[13] (Fig. 3.3c-d, Ensemble average each). These results demonstrate the power of our computational framework in revealing the hidden differential subcellular dynamics of actin regulators involved in the generation of heterogeneous morphodynamic phenotypes.

3.3.5 VASP recruitment correlates with protrusion velocity

To quantitatively assess the coordination between protrusion velocities and the dynamics of actin regulators, we performed a time-lag correlation analysis by calculating Pearson's correlation coefficients between protrusion velocities and actin regulator intensities with varying time lags in the same windows and averaged over different sampling windows (Fig. 3.1d). For actin and Arp3, the significant but relatively weak correlations were identified between the protrusion velocity and the intensities in all clusters (Fig. 3.4a-b). Conversely, the correlation of VASP in all clusters was stronger, particularly the correlation in Cluster III being the strongest in all clusters (Fig. 3.4c). Consistent with the results of cytoplasmic dynamics (Fig. 3.3e), HaloTag-TMR intensities were negatively correlated with protrusion velocities (Fig. 3.4d). Furthermore, a comparison of the maximum correlations in each cluster showed that VASP exhibited significantly stronger correlations than the Arp2/3 complex in all clusters (Fig. 3.4e, p-values in Supplementary Table 3, two-tailed Kolmogorov-Smirnov (K-S) test). These findings suggest that VASP may play a more direct role in mediating protrusion velocities in all clusters than Arp2/3.

Although the above-described conventional time correlation analysis effectively demonstrated the overall correlation between molecular dynamics and the protrusion velocity, its ability to reveal changes in this correlation over time as the protrusion progresses is limited. In other words, the correlation between the protrusion velocities and the fluorescence intensities for each specific time point was not examined in the previous analyses (Fig. 3.4a-d). Therefore, we performed sample-based correlation analyses whereby calculating pairwise Pearson correlation coefficients, $c(\{V\}_{t_i}, \{I\}_{t_j})$, between the sample of the protrusion velocity, $\{V\}_{t_i}$, at the registered time, t_i, and the sample of the actin regulator intensity $\{I\}_{t_i}$, at the registered time, t_j, over the entire probing window population (Fig. 3.1e)[13]. Then, the statistical significance of the correlations were tested by Benjamini-Hochberg multiple testing[28].

As expected, the pairwise time correlation analysis between the actin intensities and protrusion velocities (Fig. 3.4f) further supported the proposition that accelerating protrusions are mediated by the elongation of pre-existing actin filaments, whereas actin nucleation is responsible for non-accelerating protrusions. The significant regions (the black boundaries in Fig. 3.4f) of instantaneous positive correlations between the actin intensities and protrusion velocities at the leading edge found in Clusters I and II-1/2/3 (Fig. 3.4f, Cluster I, II-1/2/3) were absent in Cluster III (Fig. 3.4f, Cluster III). Notably, in the previous time lag correlation analysis, the weak correlation for actin persisted in Cluster III (Fig. 3.4a). This finding suggests that pairwise correlations at specific time points can effectively and more precisely reveal the various aspects of the coordination between protrusion velocities and the underlying molecular dynamics.

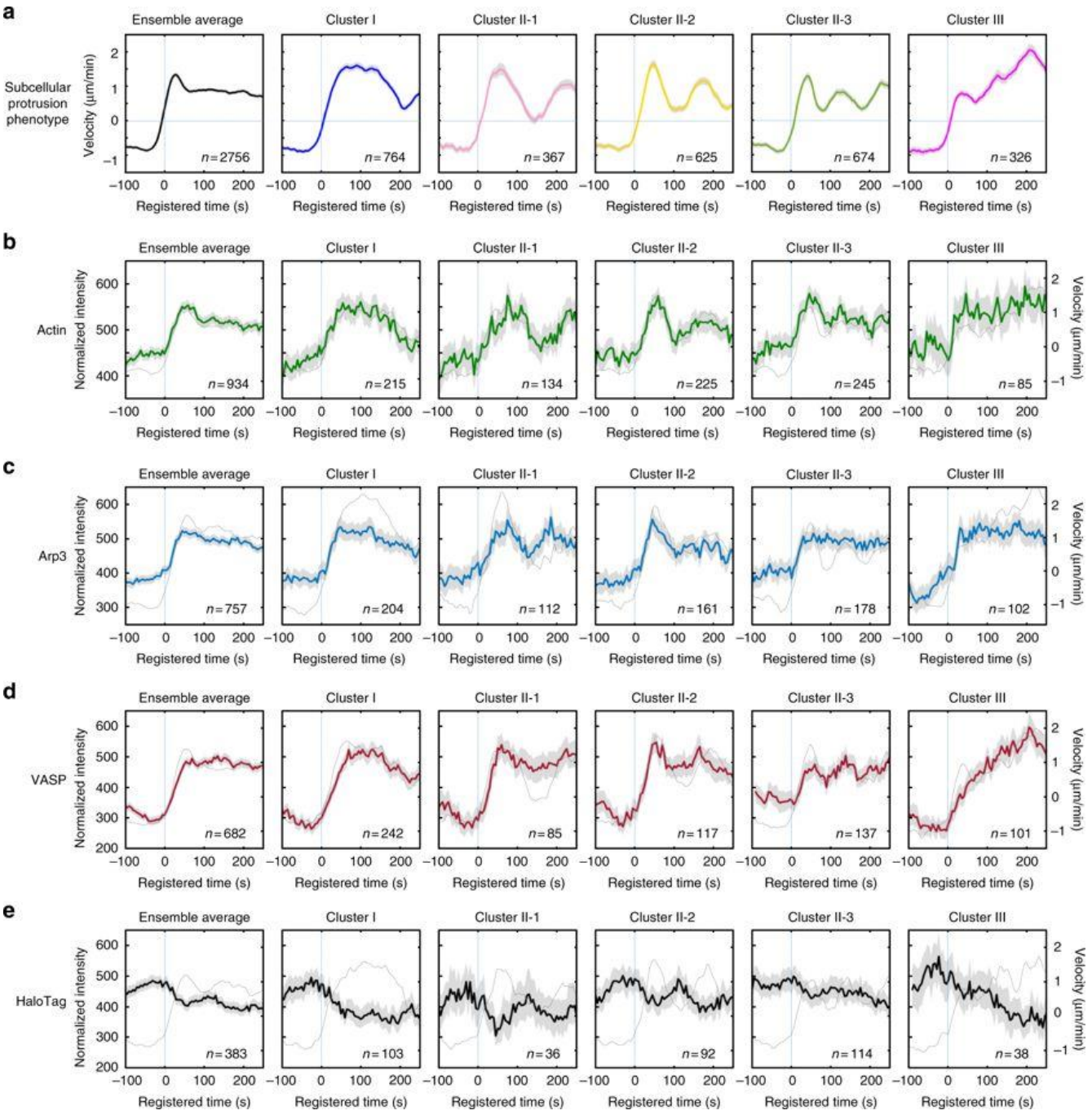


Fig. 3.3 Distinctive actin regulator dynamics associated with subcellular protrusion phenotypes.

Intriguingly, we did not identify a similarly significant instantaneous correlation between the protrusion velocity and Arp3 in any cluster (Fig. 3.4g). Conversely, we identified a significantly stronger instantaneous correlation between the VASP intensities and protrusion velocities in all clusters in the time-specific correlation analysis (Fig. 3.4h). This is consistent with the previous study such that the edge velocity and lamellipodial VASP intensity were highly correlated when the leading edges of B16 melanoma cells had a uniform rate of protrusion[29]; but our study provided substantial quantitative evidence from the samples exhibiting highly heterogeneous and non-stationary edge movements. This

further suggests that VASP compared to Arp2/3 plays a more direct role in controlling the protrusion velocity at the leading edge in all protrusion clusters. In Cluster I and II-1/2/3, VASP-dependent actin elongation likely tightly coordinates with Arp2/3 complex-mediated actin nucleation because actin exhibited a strong instantaneous correlation with protrusion velocity. Conversely, the significant and strong instantaneous correlation between VASP and the protrusion velocity in Cluster III begins to appear 100 seconds after protrusion onset (Fig. 3.4h, Cluster III), along with no correlation between actin and the protrusion velocity (Fig. 3.4f, Cluster III). This suggests that actin elongation by VASP plays a key role in the late phase of accelerating protrusion while Arp2/3 still plays a role in the early phase (Fig. 3.4i, Supplementary Note 2). We also demonstrated that VASP intensities contained sufficient information to predict protrusion phenotypes by the classification analysis (Fig. 3.4j-k, Supplementary Note 3). Notably, both the strong correlation between VASP and the protrusion velocity observed in all clusters and the postulated mode of VASP in regulating accelerating protrusions suggest that VASP plays a more critical role in generating differential protrusion phenotypes. The differences in how VASP and Arp2/3 polymerize actin further validate our interpretation. VASP facilitates actin filament elongation by binding to the barbed ends of actin filaments at the leading edge[30-32], whereas Arp2/3 binds to the sides of the mother filaments and initiates actin nucleation. Thus, the ability of Arp2/3 to directly control bared end elongation is limited[33]. Because actin elongation at the barbed end pushes the plasma membrane and generates protrusion velocity, the strong correlation between VASP activity and protrusion velocity at the leading edge is plausible.

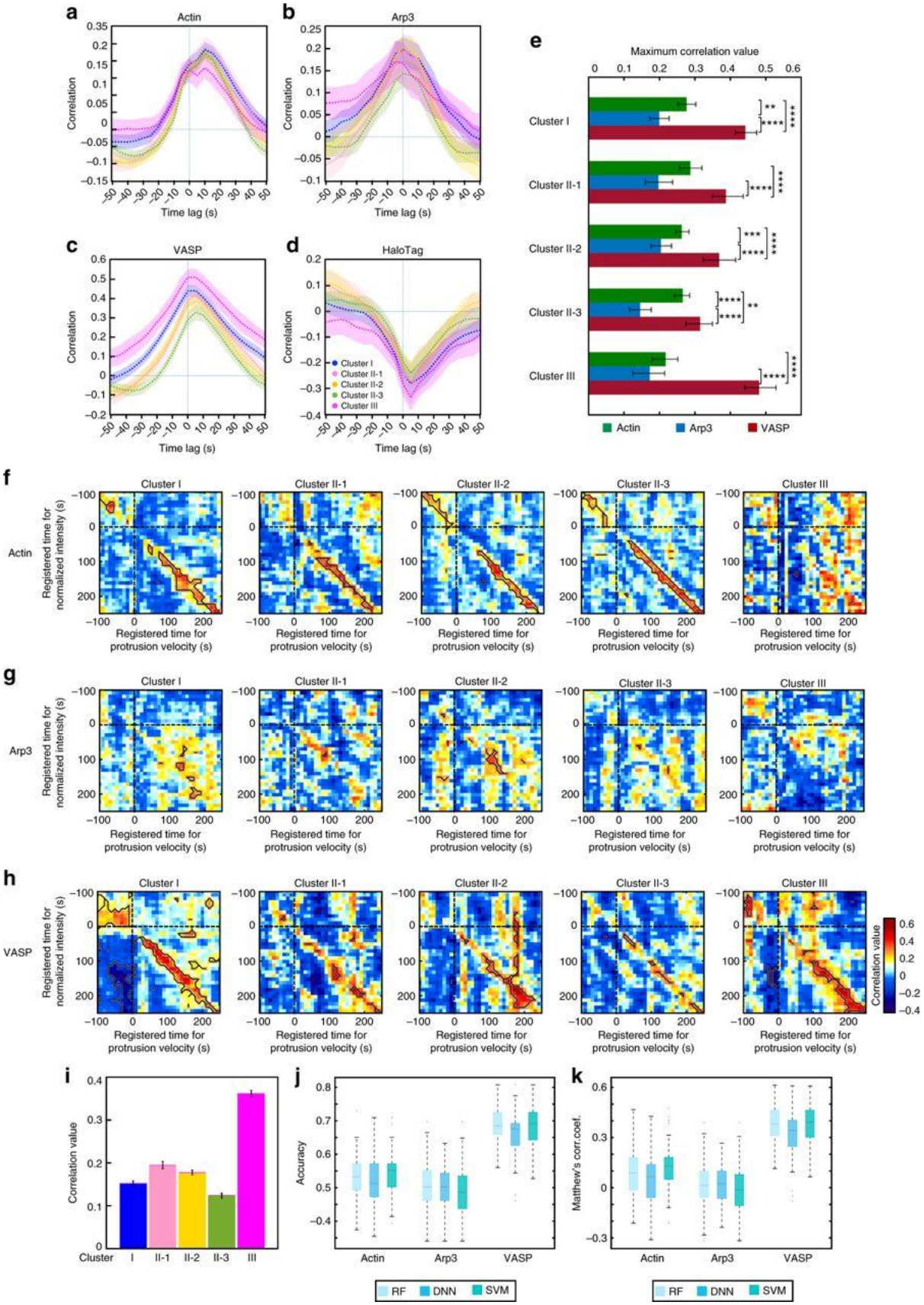


Fig. 3.4 Correlation and classification analyses between protrusion velocity and regulator dynamics.

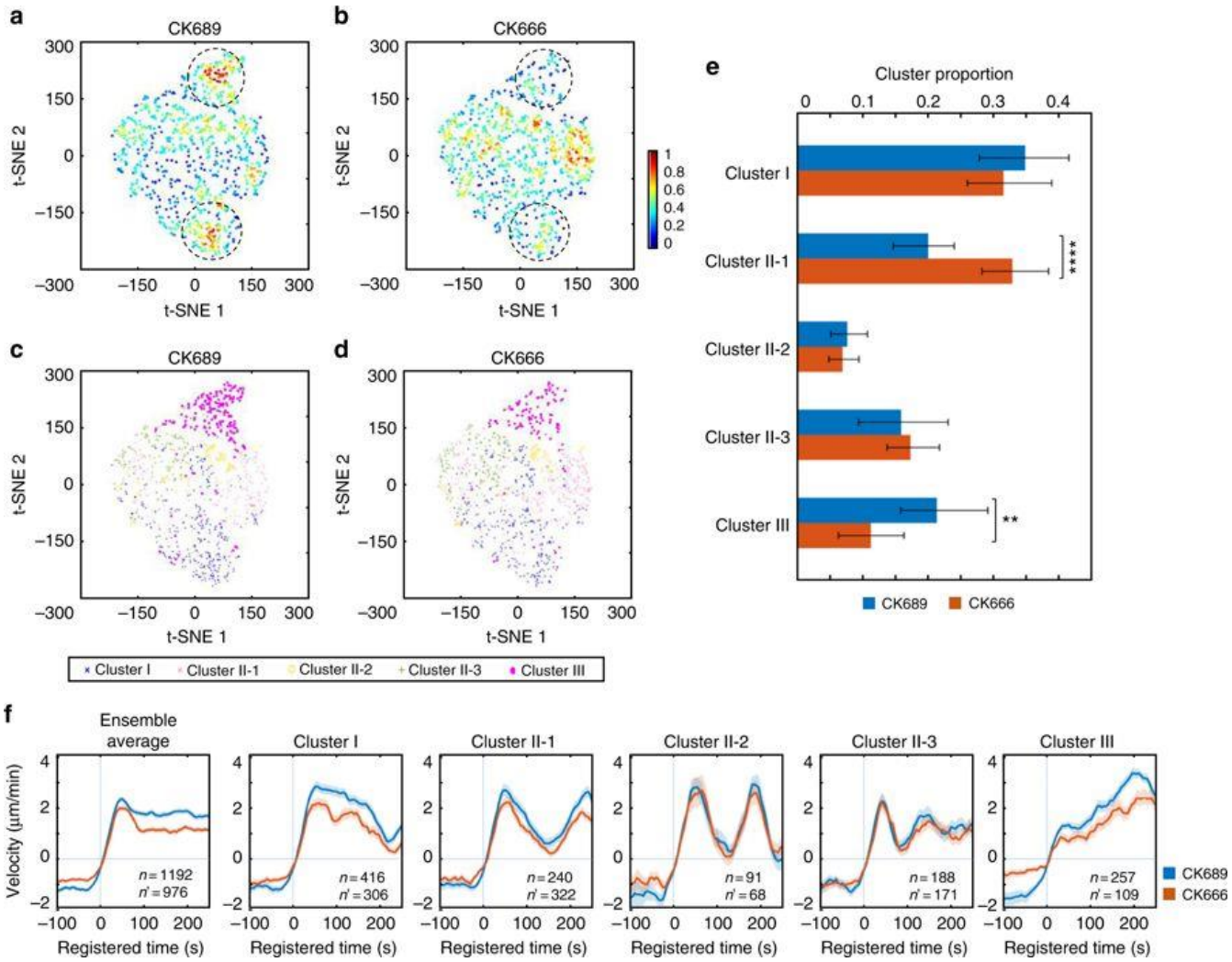


Fig. 3.5 Functional validation by pharmacological perturbation of Arp2/3.

3.3.6 Deconvolution of heterogeneous drug responses in protrusion

Our statistical analyses thus far suggest that the early recruitment of Arp2/3 at the leading edge leads to VASP recruitment to barbed ends of actin filaments, giving rise to accelerating cell protrusion. Since Arp2/3 was implicated in the early phase of accelerating protrusion, we treated PtK1 cells with an Arp2/3-specific inhibitor, CK666[34] (25 mM) to validate the functional role of Arp2/3. Notably, CK666 treated cells still exhibited highly active protrusion activities with 25 mM concentration, and they were visually indistinguishable from the control cells treated with the inactive compound, CK689. After pooling CK666 and CK689 data together, we performed the time series clustering analysis. CK666 and CK689 treated cells still exhibited similar temporal patterns in all clusters (Fig. 3.5f, Supplementary Fig. 3.6), even if the protrusion velocities in Cluster I, II-1, and III were modestly reduced by CK666 (Fig. 3.5f, Cluster I/II-1/III). The t-SNE visualization of the autocorrelation functions (ACFs) of all protrusion time series revealed that CK666 (Fig. 3.5b) affected two densely populated areas in the control (CK689) cells (the dotted circles in Fig. 3.5a-b), and overlaying the cluster assignment in these t-SNE plots revealed that Cluster III was

reduced by CK666 (Fig. 3.5c-d). The quantification of the proportion of each cluster confirmed that Cluster III was significantly reduced by the CK666 treatment (Fig. 3.5e, p = 0.0059, bootstrap sampling). In turn, this led to the significant increase of Cluster II-1 (Fig. 3.5e, p = 0.0001, bootstrap sampling). Intriguingly, the other clusters were not significantly affected by CK666, suggesting that the reduced Arp2/3 activities could be compensated by other actin regulators[35]. These results verify that Arp2/3 plays a specific functional role in accelerating protrusion. Furthermore, these demonstrate that our HACKS framework enables us to identify the susceptible clusters, which respond specifically to pharmacological perturbations.

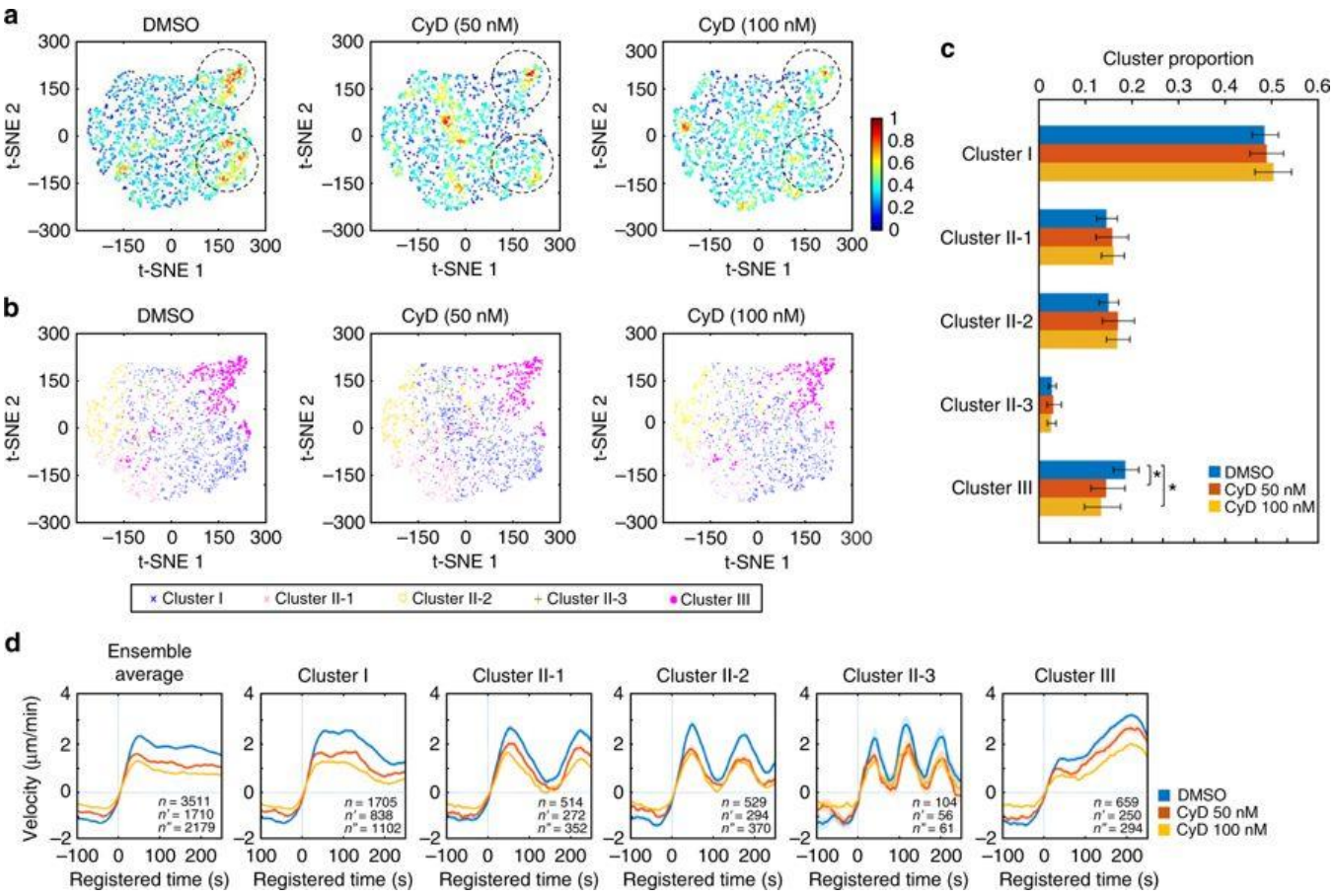


Fig. 3.6 Functional validation by pharmacological perturbation of VASP.

Next, to validate the functional role of VASP in accelerating protrusion (Cluster III), we treated PtK1 cells with low concentrations (50 and 100 nM) of Cytochalasin D (CyD) to displace VASP from the barbed ends of actin filaments [36-39]. Using immunofluorescence, we confirmed that the CyD treatment effectively removed the phosphorylated VASP, which is a functional form of VASP, from the lamellipodial leading edge of PtK1 cells (Supplementary Fig. 3.7). Consistent with our previous correlation analyses where VASP intensities correlated with protrusion velocities in all clusters, the time series clustering analysis using the pooled DMSO and CyD (50, 100nM) data revealed that protrusion velocities in all protrusion clusters in the CyD treated cells were significantly reduced in a dose-dependent manner in comparison

to DMSO treated cells (Fig. 3.6d, Supplementary Fig. 3.8). Nonetheless, the CyD treated cells retained similar clustering structures, demonstrating the specificity of the CyD treatment in these low concentrations. The t-SNE plots of ACFs of each velocity time series also revealed that two dense areas were affected by the CyD treatment (the dotted circles in Fig. 3.6a), which includes the region of Cluster III (Fig. 3.6b). The proportion of Cluster III was significantly but modestly reduced by the CyD treatment (Fig. 3.6c, $p = 0.043$ for 50 nM, 0.018 for 100 nM, bootstrap sampling).

We observed CyD treatment tended to reduce the overall protrusion velocities. Therefore, we visualized the data distributions using t-SNE with denoised protrusion velocities instead of ACFs to further investigate the effects of CyD on Cluster III in terms of regulation of protrusion velocity. This t-SNE analysis revealed high-density regions of the subcellular protrusion velocities which are highly susceptible to the CyD and CK666 treatment (the dotted circles in Fig. 3.7a-b). Overlaying the cluster assignments in these t-SNE plots showed that Cluster III contained a substantial portion of the CyD and CK666-susceptible regions (Supplementary Fig. 3.9a). Intriguingly, the tSNE plots of Cluster III of the control cells (Supplementary Fig. 3.9b) suggest that Cluster III can be largely grouped into two, which may have differential susceptibilities to CyD and CK666. Therefore, we further divided Cluster III into two sub-clusters (Fig. 3.7c-d) based on denoised protrusion velocities pooled from CyD and CK666 dataset by a community detection algorithm[40] (Supplementary Fig. 3.9c-e). While both Cluster III-1 and III-2 (Fig. 3.7g-h) maintained similar temporal patterns, Cluster III-2 had substantially stronger accelerating activities compared to Cluster III-1 (Fig. 3.7g, DMSO (Cluster III-1)/DMSO (Cluster III-2) and Fig. 3.7h, CK689 (Cluster III-1)/CK689 (Cluster III-2). Intriguingly, the t-SNE plots revealed that ‘strongly accelerating protrusion’ (Cluster III-2) was preferentially affected by the CyD (Fig. 3.7c) and CK666 (Fig. 3.7d) treatment. The quantification of the proportion of these sub-clusters (Fig. 3.7e-f) confirmed that strongly accelerating protrusion (Cluster III-2) was significantly reduced by the CyD treatment in comparison to DMSO treatment in a dose-dependent manner ($p = 0.024$ for 50 nM, < 0.0001 for 100 nM, bootstrap sampling) (Fig. 3.7e, Cluster III-2), whereas the weakly accelerating protrusion (Cluster III-1) was increased ($p = 0.006$ for 100 nM, , bootstrap sampling) (Fig. 3.7e, Cluster III-1). Therefore, the average protrusion velocities in Cluster III in CyD treatment were significantly reduced to be comparable to Cluster III-1 in DMSO treatment and was significantly lower than Cluster III-2 (Fig. 3.7g). Consistently, the proportion of strongly accelerating protrusion (Cluster III-2) was significantly reduced by the CK666 treatment (Fig. 3.7f, $p = 0.0026$, bootstrap sampling) and the average velocities of Cluster III in CK666 treatment were also reduced to those of weakly accelerating protrusion (Cluster III-1) in CK689 treatment (Fig. 3.7h). These data demonstrate HACKS allowed us to successfully identify the drug-susceptible sub-phenotypes, where strongly accelerating protrusion is specifically affected by Arp2/3 and VASP inhibition.

Next, we further investigated whether dynamics of VASP and Arp3 in accelerating protrusion is differentially regulated between Cluster III-1 and Cluster III-2. We divided the intensity time series of

VASP and Arp3 in Cluster III (Fig. 3.7j and l) into two sub-clusters and compared their differential dynamics. The recruitment dynamics of VASP in Cluster III-2 exhibited strong increase, while that of Cluster III-1 exhibited only moderate elevation, which is within the 95% confidence interval of the mean (Fig. 3.7j). On the other hand, Arp3 intensity patterns in Cluster III-1 and 2 were almost identical (Fig. 3.7l). This is consistent with our notion that Arp2/3 is involved in initiating accelerating protrusion and VASP is important in the output of accelerating protrusion. To functionally confirm this, we compared Arp3-GFP fluorescence dynamics at the leading edges in each cluster without and with 100 nM CyD treatment (Fig. 3.7m and n, Supplementary Fig. 3.10). To this end, we normalized Arp3 intensities at the leading edge by those of the lamella region in the same cell to quantitatively compare the Arp3 accumulation in different experimental condition. Under CyD treatment, the Arp3 fluorescence normalized by lamella intensity still started to increase at the protrusion onset in Cluster III (Fig. 3.7n). Normalized Arp3 fluorescence continued to increase up to 4-fold more than the DMSO control while the protrusion velocity did not increase (Fig. 3.7m). First, this suggests that CyD treatment did not affect the initial Arp2/3 recruitment to the leading edge in accelerating protrusion, which proposes that Arp2/3 precedes VASP in accelerating protrusion. In addition, this data shows that even increasing Arp2/3 recruitment under CyD treatment could not produce strongly accelerating protrusion without VASP activity. Therefore, the specific temporal coordination between Arp2/3 and VASP is crucial to the strongly accelerating protrusion. Notably, such molecular temporal coordination was reported to be involved in cell protrusion[12,13,41,42]. Particularly, PI3K has been known to increase after protrusion onset to stabilize nascent cell protrusion[41]. Taken together, our HACKS framework combined with pharmacological perturbations effectively demonstrated that heterogeneous edge movements could be deconvolved into variable protrusion phenotypes to reveal the underlying differential regulation of actin molecular dynamics. We also successfully demonstrated that we could monitor the changes in actin regulator dynamics induced by functional perturbation.

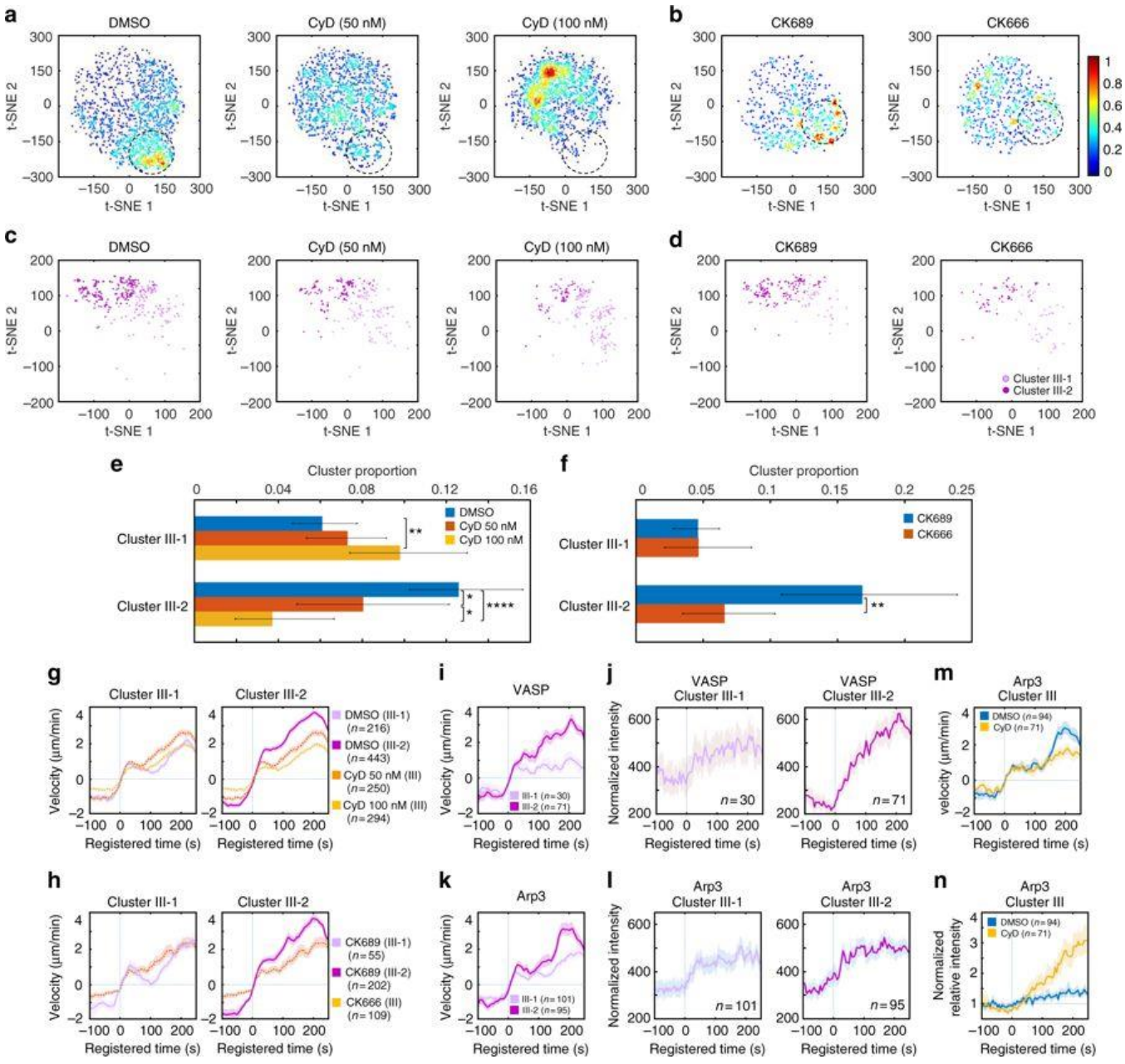


Fig. 3.7 Functional validation of the differential coordination between Arp2/3 and VASP in strong accelerating protrusion.

3.4 Discussion

We have demonstrated that our computational framework HACKS could effectively deconvolve heterogeneous subcellular protrusion activities into distinct protrusion phenotypes, establish an association between each protrusion phenotype and the underlying differential actin regulator dynamics, and reveal specific phenotypes susceptible to pharmacological perturbations. Although previous studies have examined the spatiotemporal patterning of cell edge dynamics[11,43-45], our study is the first to propose an effective framework to analyze the temporal heterogeneity in protrusion activities at the subcellular level and identify the subcellular protrusion phenotype. Together with the functional assays, we identified

‘strongly accelerating protrusion’ susceptible to the pharmacological perturbations. Although previous studies also described persistent protrusion based on protrusion distance on a longer time scales[11,13,37,46], we first dissected protrusion phenotypes at fine spatiotemporal scales and quantitatively characterize persistently ‘accelerating protrusion’. Intriguingly, accelerating protrusion was later shown to be regulated by differential mechanisms, although they accounted for a minor portion of entire sampled protrusions. This finding indicates that identifying even a small subset of phenotypes is crucial to fully understand the mechanism underlying heterogeneous cellular behaviors.

We were also able to quantitatively measure how the underlying molecular dynamics are coordinated with protrusion phenotypes, thereby revealing the hidden variability of molecular regulatory mechanisms. Elucidating precise differential regulatory mechanisms related to protrusion heterogeneity has been difficult partly because it remains challenging to experimentally perturb a subset of molecules involved with specific subcellular phenotypes *in situ*. To address this challenge, our framework employed highly specific computational analyses. The result of our analyses provided quantitative and detailed information about the differential coordination between molecular dynamics and the protrusion phenotype at the subcellular level.

We also demonstrated that we could deconvolve the heterogeneity of drug responses of cellular protrusions using our HACKS framework by mapping protrusion time-series to two-dimensional phenotypic space using t-SNE and our time series clustering results. This approach revealed specific protrusion phenotypes which are most susceptible to pharmacological perturbations and functionally validated our hypothesis drawn from the statistical analysis that the temporal coordination between Arp2/3 and VASP drives the accelerating protrusion. To date, the Arp2/3 complex has been widely accepted as a master organizer of branched actin networks in lamellipodia that acts by nucleating actin filaments[27], whereas VASP has been thought to be an elongator of actin filaments or anti-capper of the barbed ends[30,31,35,47]. In this study, we focused on the distinct recruitment dynamics of Arp3 and VASP identified in the accelerating protrusion phenotype (Cluster III). This suggested that Arp2/3-dependent actin nucleation provides a branched structural foundation for protrusion activity, and VASP-mediated actin elongation subsequently takes over to persistently accelerate protrusions. Our functional studies using CK666 and Cytochalasin D confirmed that this coordination is critical to strongly accelerating cell protrusion and the recruitment timing and duration of Arp3 and VASP is finely regulated to generate differential protrusion activities. Notably, VASP was reported to increase cell protrusion activities[37,38,46], and has been implicated in cancer invasion and migration[37,48,49]. Thus, the coordination of Arp2/3 and VASP may regulate the plasticity of protrusion phenotypes, and the functional deregulation of the VASP or its isoforms in cancer may promote cellular migratory behaviors by promoting accelerating protrusion.

Furthermore, we consider HACKS is not limited to the analyses of subcellular protrusion heterogeneity: we anticipate that it can be expanded to study the morphodynamic heterogeneity of other types of

cytoskeletal structures and membrane-bound organelles. Together with the further development of unsupervised learning along with an increased repertoire of molecular dynamics, we expect our machine learning framework to accelerate the mechanistic understanding of heterogeneous cellular and subcellular behaviors.

Figure Legends

Fig 3.1 Schematic representation of the analytical steps of HACKS. **a** Fluorescence time-lapse movies of the leading edge of a migrating PtK1 cell expressing flourescent-tagged proteins of interest (an Arp3-HaloTag expressing cell is presented here) was taken at 5 s per frame, and then probing windows (500 by 500 nm) are generated to track the cell edge movement and sample protrusion velocities and fluorescence intensities. **b** The protrusion distance is registered with respect to protrusion onsets ($t = 0$). Time series of protrusion velocities are then aligned. **c** The protrusion phenotypes are identified by a time series clustering analysis and associated with actin regulator dynamics. **d**–**f** Correlation analysis between time series of the protrusion velocities and fluorescence intensities. Schematic diagrams of time-lag (**d**) and time-specific correlation analysis (**e**) are presented. Classification analysis is performed to computationally validate the result by predicting protrusion phenotypes based on molecular dynamics. **g** The hypotheses drawn from the computational analysis are functionally validated by drug tests. The phenotypes susceptible to pharmacological perturbations are identified based on t-SNE plots. The drug-sensitive phenotypes are further analyzed by quantifying the drug effects on cluster proportion and the associated molecular dynamics

Fig. 3.2 Subcellular protrusion phenotypes revealed by a time series clustering analysis. a–**c** A representative cluster assignment on a time-lapse movie of a PtK1 cell stained with CellMask DeepRed. Edge evolution on 5 s interval (**a**), cluster assignments of each probing window on every four frames (20 s interval) (**b**), and the space-time maps of instantaneous edge velocity (**c**) of the entire cell edge and the indicated cluster regions. Scale bar: 10 μm. **d** Raw velocity maps for Cluster I, II-1, II-2, II-3, and III. All time series are registered with respect to protrusion onset ($t = 0$). **e** Average time series of protrusion velocity registered at protrusion onsets ($t = 0$) in each cluster. Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling. *n* indicates the number of time series in each cluster. The time lapse movies of 36 cells were used in this analysis. **f** Proportions of each cluster in entire samples or individual cells expressing fluorescent actin, Arp3, VASP, and HaloTag, respectively. **g** Decision graph of the density peak clustering analysis of protrusion velocities. **h** A t-SNE plot of the autocorrelation functions of protrusion velocity time series overlaid with cluster assignments. **i** Spatial conditional distribution of each cluster. Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling.

Fig. 3.3 Distinctive actin regulator dynamics associated with subcellular protrusion phenotypes. a Ensemble averaged velocity time series of entire samples and averaged velocity time series sampled in each cluster. All time series are registered with respect to protrusion onset ($t = 0$). **b**–**e** Ensemble averaged normalized fluorescence intensity time series of entire samples and normalized fluorescence intensity time series in each cluster. All time series are registered with respect to protrusion onset ($t = 0$). Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling. The dotted lines in **b**-**e** indicate protrusion velocity time series associated with the indicated fluorescent proteins. *n* indicates the number of time series sampled in each cluster. The numbers of cells used for the analyses are 36 (**a**), 10 (**b**), 11 (**c**), 9 (**d**) and 6 (**e**) respectively. The number of time series sampled and the number of cells imaged for each cluster is summarized in Supplementary Table 1.

Fig. 3.4 Correlation and classification analyses between protrusion velocity and actin regulator dynamics. **a**–**d** Time-lag correlation analysis based on Pearson's cross-correlation of edge velocity and actin (**a**), Arp3 (**b**), VASP (**c**), and HaloTag (**d**). Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling. The number of samples used for the analysis is identical from Fig.3b–e. **e** Comparison and statistical testing of maximum correlation coefficients from **a**–**d** in each cluster. The error bar indicates 95% confidence interval of the mean by bootstrapping. $^{**}p < 0.01$, $^{***}p < 0.001$ and $^{****}p < 0.0001$ indicate the statistical significance by two-tailed two-sample Kolmogorov–Smirnov (KS) test. The *p*-values are listed in Supplementary Table 3. **f**–**h** Time-specific correlation analysis based on pairwise Pearson's correlation coefficients of protrusion velocity and fluorescence intensity time series registered relative to protrusion onset. The regions surrounded by the black lines are statistically significant correlation by Benjamini-Hochberg multiple hypothesis testing. **i** Pearson's correlation coefficients between early Arp3 intensities and late protrusion velocities in each cluster. The error bar indicates 95% confidence interval of the mean by bootstrapping. The numbers of samples in this analysis are 204 (Cluster I), 112 (Cluster II-1), 161 (Cluster II-2), 178 (Cluster II-3) and 102 (Cluster III) respectively. **j**–**k** Classification analysis of Cluster III against Clusters I/II based on fluorescent intensity time series. Boxplots of the accuracy (**j**) and Matthews correlation coefficients (**k**) represent multiple classification results. RF stands for Random Forest, DNN for Deep Neural Network, and SVM for Support Vector Machine. The central line indicates median, and both edges of the box each represent 25^{th} and 75^{th} percentiles. The numbers of samples used in these analyses are 934 (actin), 757 (Arp3) and 682 (VASP) respectively.

Fig. 3.5 Functional validation by pharmacological perturbation of Arp2/3. a–**d** t-SNE plots of autocorrelation functions of protrusion velocity time series overlaid with the density of data (**a**, **b**) and cluster assignments (**c**, **d**). **e** Comparison of the proportion of each cluster between CK689 (50 μM, inactive control compound) and CK666 (50 μM). The error bars indicate 95% confidence interval of

the mean of the cluster proportions. $^{**}p < 0.01$ and $^{****}p < 0.0001$ indicate the statistical significance by bootstrap sampling. **f** Ensemble averaged velocity time series of entire samples and averaged velocity time series sampled in each cluster in CK689 or CK666-treated cells. All time series are registered with respect to protrusion onset ($t = 0$). Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling. n and n' indicate the number of time series sampled in each cluster for CK689 and CK666, respectively. The numbers of cells used in the analysis are both 10 (CK689 and CK666).

Fig. 3.6 Functional validation by pharmacological perturbation of VASP. a, **b** t-SNE plots of autocorrelation functions of protrusion velocity time series overlaid with the density of data (**a**) and cluster assignments (**b**). **c** Dose-response of the proportions of clusters to CyD. The error bars indicate 95% confidence interval of the mean of the cluster proportions. $^{*}p < 0.05$ indicates the statistical significance by bootstrap sampling. **d** Ensemble averaged velocity time series of entire samples and averaged velocity time series sampled in each cluster in DMSO or Cytochalasin D (50 or 100 nM)-treated cells. All time series are registered with respect to protrusion onset ($t = 0$). Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling. n, n' and n'' indicate the number of time series sampled in each cluster for DMSO, CyD 50nM, and CyD 100nM, respectively. The numbers of cells used in these analyses are 22 (DMSO), 16 (CyD 50 nM) and 20 (CyD 100 nM), respectively.

Fig. 3.7 Functional validation of the differential coordination between Arp2/3 and VASP in strong accelerating protrusion. a, **b** t-SNE plots of the denoised protrusion velocity time series of the whole sample overlaid with the density of data. **c**, **d** t-SNE plots of the denoised velocities of the sub-clusters (Cluster III-1 and III-2) in Cluster III. **e**, **f** Comparison of the proportion of Cluster III-1 and III-2 upon Cytochalasin D treatment (**e**) or CK666 treatment (**f**). The error bars indicate 95% confidence interval of the mean of the cluster proportions. $^{*}p < 0.05$, $^{**}p < 0.01$, and $^{****}p < 0.0001$ indicate the statistical significance by bootstrap sampling. The numbers of cells used in this analysis are 22 (DMSO), 16 (CyD 50 nM), 20 (CyD 100 nM) and 10 (both CK689 and CK666), respectively. **g**, **h** Averaged velocity time series sampled in Cluster III (Cytochalasin D (**g**) or CK666 (**h**)), Cluster III-1 and Cluster III-2 (DMSO (**g**) or CK689 (**h**)). **i**–**l** Averaged velocity time series and normalized fluorescence intensity time series of Cluster III-1 and Cluster III-2 from unperturbed VASP-expressing cells (**i**, **j**) or Arp3-expressing cells (**k**, **l**). **m**, **n** Averaged velocity time series (**m**) and normalized fluorescence intensity time series of Cluster III (**n**) in Arp3-expressing cells upon treatment of DMSO or Cytochalasin D (100 nM). All time series are registered with respect to protrusion onset ($t = 0$). Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals of the mean computed by bootstrap sampling. n indicates the number of time series in each cluster.

Literature Cited

1 Small, J. V., Stradal, T., Vignal, E. & Rottner, K. The lamellipodium: where motility begins. *Trends Cell Biol* 12, 112-120 (2002).
2 Pankov, R. *et al.* A Rac switch regulates random versus directionally persistent cell migration. *J Cell Biol* 170, 793-802, doi:10.1083/jcb.200503152 (2005).
3 Lauffenburger, D. A. & Horwitz, A. F. Cell migration: a physically integrated molecular process. *Cell* 84, 359-369 (1996).
4 Guirguis, R., Margulies, I., Taraboletti, G., Schiffmann, E. & Liotta, L. Cytokine-induced pseudopodial protrusion is coupled to tumour cell migration. *Nature* 329, 261-263, doi:10.1038/329261a0 (1987).
5 Morikawa, Y. *et al.* Actin cytoskeletal remodeling with protrusion formation is essential for heart regeneration in Hippo-deficient mice. *Sci Signal* 8, ra41, doi:10.1126/scisignal.2005781 (2015).
6 Antonello, Z. A., Reiff, T., Ballesta-Illan, E. & Dominguez, M. Robust intestinal homeostasis relies on cellular plasticity in enteroblasts mediated by miR-8-Escargot switch. *EMBO J* 34, 2025-2041, doi:10.15252/embj.201591517 (2015).
7 Liu, Y. H. *et al.* Protrusion-localized STAT3 mRNA promotes metastasis of highly metastatic hepatocellular carcinoma cells in vitro. *Acta Pharmacol Sin* 37, 805-813, doi:10.1038/aps.2015.166 (2016).
8 Taniuchi, K., Furihata, M., Hanazaki, K., Saito, M. & Saibara, T. IGF2BP3-mediated translation in cell protrusions promotes cell invasiveness and metastasis of pancreatic cancer. *Oncotarget* 5, 6832-6845, doi:10.18632/oncotarget.2257 (2014).
9 Ioannou, M. S. *et al.* DENND2B activates Rab13 at the leading edge of migrating cells and promotes metastatic behavior. *J Cell Biol* 208, 629-648, doi:10.1083/jcb.201407068 (2015).
10 Leithner, A. *et al.* Diversified actin protrusions promote environmental exploration but are dispensable for locomotion of leukocytes. *Nat Cell Biol* 18, 1253-1259, doi:10.1038/ncb3426 (2016).
11 Machacek, M. & Danuser, G. Morphodynamic profiling of protrusion phenotypes. *Biophys J* 90, 1439-1452, doi:10.1529/biophysj.105.070383 (2006).
12 Machacek, M. *et al.* Coordination of Rho GTPase activities during cell protrusion. *Nature* 461, 99-103, doi:10.1038/nature08242 (2009).
13 Lee, K. *et al.* Functional hierarchy of redundant actin assembly factors revealed by fine-grained registration of intrinsic image fluctuations. *Cell Syst* 1, 37-50, doi:10.1016/j.cels.2015.07.001 (2015).
14 Altschuler, S. J. & Wu, L. F. Cellular heterogeneity: do differences make a difference? *Cell* 141, 559-563, doi:10.1016/j.cell.2010.04.033 (2010).
15 Raj, A. & van Oudenaarden, A. Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell* 135, 216-226, doi:10.1016/j.cell.2008.09.050 (2008).
16 Slack, M. D., Martinez, E. D., Wu, L. F. & Altschuler, S. J. Characterizing heterogeneous cellular responses to perturbations. *Proc Natl Acad Sci U S A* 105, 19306-19311, doi:10.1073/pnas.0807038105 (2008).
17 Levine, J. H. *et al.* Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. *Cell* 162, 184-197, doi:10.1016/j.cell.2015.05.047 (2015).
18 Patel, A. P. *et al.* Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* 344, 1396-1401, doi:10.1126/science.1254257 (2014).
19 Shafqat-Abbasi, H. *et al.* An analysis toolbox to explore mesenchymal migration heterogeneity reveals adaptive switching between distinct modes. *Elife* 5, e11384, doi:10.7554/eLife.11384 (2016).
20 Sailem, H., Bousgouni, V., Cooper, S. & Bakal, C. Cross-talk between Rho and Rac GTPases drives deterministic exploration of cellular shape space and morphological heterogeneity. *Open Biol* 4, 130132, doi:10.1098/rsob.130132 (2014).
21 Huang, N. E. *et al.* The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 454 (1998).

22 Keogh, E., Lin, J. & Fu, A. HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. *In Proc. of the 5th IEEE International Conference on Data Mining* 226 - 233 (2005).
23 Rodriguez, A. & Laio, A. Machine learning. Clustering by fast search and find of density peaks. *Science* 344, 1492-1496, doi:10.1126/science.1242072 (2014).
24 Davies, D. L. & Bouldin, D. W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, 224-227 (1979).
25 Caliński, T. & Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3, 1-27 (1974).
26 Los, G. V. *et al.* HaloTag: a novel protein labeling technology for cell imaging and protein analysis. *ACS Chem Biol* 3, 373-382, doi:10.1021/cb800025k (2008).
27 Pollard, T. D. & Borisy, G. G. Cellular motility driven by assembly and disassembly of actin filaments. *Cell* 112, 453-465 (2003).
28 Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, 289-300 (1995).
29 Rottner, K., Behrendt, B., Small, J. V. & Wehland, J. VASP dynamics during lamellipodia protrusion. *Nat Cell Biol* 1, 321-322, doi:10.1038/13040 (1999).
30 Barzik, M. *et al.* Ena/VASP proteins enhance actin polymerization in the presence of barbed end capping proteins. *J Biol Chem* 280, 28653-28662, doi:10.1074/jbc.M503957200 (2005).
31 Breitsprecher, D. *et al.* Clustering of VASP actively drives processive, WH2 domain-mediated actin filament elongation. *EMBO J* 27, 2943-2954, doi:10.1038/emboj.2008.211 (2008).
32 Hansen, S. D. & Mullins, R. D. Lamellipodin promotes actin assembly by clustering Ena/VASP proteins and tethering them to actin filaments. *Elife* 4, doi:10.7554/eLife.06585 (2015).
33 Machesky, L. M. *et al.* Scar, a WASp-related protein, activates nucleation of actin filaments by the Arp2/3 complex. *Proc Natl Acad Sci U S A* 96, 3739-3744 (1999).
34 Nolen, B. J. *et al.* Characterization of two classes of small molecule inhibitors of Arp2/3 complex. *Nature* 460, 1031-1034, doi:10.1038/nature08231 (2009).
35 Rotty, J. D. *et al.* Profilin-1 serves as a gatekeeper for actin assembly by Arp2/3-dependent and -independent pathways. *Dev Cell* 32, 54-67, doi:10.1016/j.devcel.2014.10.026 (2015).
36 Bear, J. E. *et al.* Antagonism between Ena/VASP proteins and actin filament capping regulates fibroblast motility. *Cell* 109, 509-521 (2002).
37 Bae, Y. H. *et al.* Profilin1 regulates PI(3,4)P2 and lamellipodin accumulation at the leading edge thus influencing motility of MDA-MB-231 cells. *Proc Natl Acad Sci U S A* 107, 21547-21552, doi:10.1073/pnas.1002309107 (2010).
38 Lacayo, C. I. *et al.* Emergence of large-scale cell morphology and movement from local actin filament growth dynamics. *PLoS Biol* 5, e233, doi:10.1371/journal.pbio.0050233 (2007).
39 Neel, N. F. *et al.* VASP is a CXCR2-interacting protein that regulates CXCR2-mediated polarization and chemotaxis. *J Cell Sci* 122, 1882-1894, doi:10.1242/jcs.039057 (2009).
40 Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci U S A* 105, 1118-1123, doi:10.1073/pnas.0706851105 (2008).
41 Welf, E. S., Ahmed, S., Johnson, H. E., Melvin, A. T. & Haugh, J. M. Migrating fibroblasts reorient directionality by a metastable, PI3K-dependent mechanism. *J Cell Biol* 197, 105-114, doi:10.1083/jcb.201108152 (2012).
42 Johnson, H. E. *et al.* F-actin bundles direct the initiation and orientation of lamellipodia through adhesion-based signaling. *J Cell Biol* 208, 443-455, doi:10.1083/jcb.201406102 (2015).
43 Martin, K. *et al.* Spatio-temporal co-ordination of RhoA, Rac1 and Cdc42 activation during prototypical edge protrusion and retraction dynamics. *Sci Rep* 6, 21901, doi:10.1038/srep21901 (2016).
44 Verkhovsky, A. B. The mechanisms of spatial and temporal patterning of cell-edge dynamics. *Curr Opin Cell Biol* 36, 113-121, doi:10.1016/j.ceb.2015.09.001 (2015).
45 Dobereiner, H. G. *et al.* Lateral membrane waves constitute a universal dynamic pattern of motile cells. *Phys Rev Lett* 97, 038102, doi:10.1103/PhysRevLett.97.038102 (2006).

46 Barnhart, E. L., Allard, J., Lou, S. S., Theriot, J. A. & Mogilner, A. Adhesion-Dependent Wave Generation in Crawling Cells. *Curr Biol* 27, 27-38, doi:10.1016/j.cub.2016.11.011 (2017).
47 Hansen, S. D. & Mullins, R. D. VASP is a processive actin polymerase that requires monomeric actin for barbed end association. *J Cell Biol* 191, 571-584, doi:10.1083/jcb.201003014 (2010).
48 Carmona, G. *et al.* Lamellipodin promotes invasive 3D cancer cell migration via regulated interactions with Ena/VASP and SCAR/WAVE. *Oncogene* 35, 5155-5169, doi:10.1038/onc.2016.47 (2016).
49 Philippar, U. *et al.* A Mena invasion isoform potentiates EGF-induced carcinoma cell invasion and metastasis. *Dev Cell* 15, 813-828, doi:10.1016/j.devcel.2008.09.003 (2008).
50 Sundar, A., Pahwa, V., Das, C., Deshmukh, M. & Robinson, N. A Comprehensive Assessment of the Performance of Modern Algorithms for Enhancement of Digital Volume Pulse Signals *International Journal of Pharma Medicine and Biological Sciences* 5, 91-98 (2016).
51 Lin, J., Keogh, E., Lonardi, S. & Chiu, B. A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM*, 2-11 (2003).
52 Pierpaolo, D. & Maharaj, E. A. Autocorrelation-based fuzzy clustering of time series. *Fuzzy Sets and Systems* 160, 3565-3589 (2009).
53 Mori, U., Mendiburu, A. & Lozano, J. A. Distance Measures for Time Series in R: The TSdist Package. (2016).
54 Marek Walesiak, A. D. clusterSim: Searching for Optimal Clustering Procedure for a Data Set. (2017).
55 Wickelmaier, F. An introduction to MDS. *Sound Quality Research Unit, Aalborg University, Denmark*, 46 (2003).
56 Laurens van der, M. & G., H. Visualizing data using t-SNE *Journal of Machine Learning Research* 9, 2579-2605 (2008).
57 Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20, 53-65 (1987).
58 Ho, T. K. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 832–844 (1998).
59 Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learning* 20, 273–297 (1995).
60 LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324 (1998).
61 Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825-2830, (2011).
62 Lematre, G., Nogueira, F. & Aridas, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 7, 1-5 (2016).

Chapter 4

DeepHACKS: Deep Learning-based Subcellular Phenotyping of Leading Edge Dynamics Reveals Fine Differential Drug Responses

Intracellular processes such as cytoskeletal organization and organelle dynamics exhibit massive subcellular heterogeneity. Although recent advances in fluorescence microscopy allow researchers to acquire an unprecedented amount of live cell image data at high spatiotemporal resolutions, the traditional ensemble-averaging of uncharacterized subcellular heterogeneity could mask important activities. Moreover, the curse of dimensionality of these complex dynamic datasets prevents access to critical mechanistic details of subcellular processes. Here, we establish an unsupervised machine learning framework called DeepHACKS (Deep phenotyping of Heterogeneous Activities in the Coordination of cytosKeleton at the Subcellular level) for "deep phenotyping," which identifies rare subcellular phenotypes specifically sensitive to molecular and environmental perturbations. DeepHACKS dissects the heterogeneity of subcellular time-series datasets by allowing bi-directional LSTM (Long-Short Term Memory) neural networks to extract fine-grained temporal features by integrating autoencoders with traditional machine learning outcomes. We applied DeepHACKS to subcellular protrusion dynamics in pharmacologically and metabolically perturbed epithelial cells, revealing fine differential responses of leading edge dynamics specific to each perturbation. Particularly, the framework revealed the emergence of rare protrusion phenotypes upon different perturbations, such as "bursting" protrusion. This suggests that the temporal features directly learned from leading edge dynamics enable fine-grained identification of drug-related subcellular phenotypes, which may not be possible from static cell images. In summary, our study provides an analytical framework for detailed and quantitative understandings of molecular mechanisms hidden in their heterogeneity. DeepHACKS can be potentially applied to analyze various time-series data measured from other subcellular processes.

4.1 Introduction

Recent advances in fluorescence microscopy allow researchers to acquire an unprecedented amount of live cell image data at high spatiotemporal resolutions[1-2]; however Intracellular processes such as cytoskeletal organization and organelle dynamics exhibit massive subcellular heterogeneity[3-5], which makes it difficult to understand these rich microscopy datasets. Moreover, the traditional ensemble-averaging of uncharacterized subcellular heterogeneity could lead to the loss of critical mechanistic details. For instance, if only a small subcellular region in a small subpopulation of cells specifically respond to perturbations, it will be easily overlooked by the conventional assays[6-7].

In cancer genomics, detecting rare mutations and cell types by deep sequencing has been critical to the understanding of tumor heterogeneity[8]. However, little effort in cell biology has been made for "deep phenotyping"[9-10], which identifies detailed and rare phenotypes of the subcellular process from

microscopy datasets. Over the last decade, conventional image software[11-13] has been facilitated the quantitative analyses of molecular and cellular events from microscopy images but has limited capacity in characterizing their heterogeneity. Unsupervised machine learning (ML)[14] is highly capable of finding relationships within complex datasets with minimal human input, making it powerful in recognizing hidden patterns. Therefore, unsupervised ML for high-resolution fluorescence time-lapse images can open an exciting new opportunity that can automatically unravel the heterogeneity of subcellular processes to identify deep subcellular phenotypes[7].

For the effective machine learning of high dimensional datasets, it is necessary to project the raw data onto low-dimensional manifolds to avoid the curse of dimensionality[14-15]. Therefore, in traditional machine learning settings, the hand-crafted features are extracted for the dimensional reduction of data. Previously, we developed an unsupervised ML pipeline[7] , which relied on the hand-crafted time-series feature, ACF (Auto-Correlation Function), to deconvolve the time series heterogeneity from live cell movies, which revealed heterogeneous responses of cell protrusion to drug perturbations[7]. Hand-crafted features, however, depend on prior knowledge and are limited in representing large complex dataset comprehensively. Moreover, increased spatiotemporal heterogeneity, it is unclear if we will be able to identify rare phenotypes using the hand-crafted features even with iterative feature engineering.

Over the last decade, deep learning (DL)[16-18] had risen to be a mainstream ML method, overcame the performance of traditional ML, and surpassed the human capabilities in many areas[19-22]. Unlike traditional ML, DL does not use hand-crafted features, but rather it learns the data representation directly from raw data[23-24]. The learned features by DL will be better positioned to capture more accurate information from complex datasets, enabling us to unravel the heterogeneity of the data in detail for deep phenotyping. One drawback of the DL approach, however, is that DL is considered to be a black-box approach where the interpretability of learned features is limited. Therefore, the outcomes may not be compatible with human intuition.

To address this challenge, we developed an unsupervised ML framework integrating traditional ML and DL, called DeepHACKS (Deep phenotyping of Heterogeneous Activities in the Coordination of cytosKeleton at the Subcellular level). DeepHACKS allows us to deconvolve the heterogeneity of subcellular time-series dataset using DL-based features integrated with traditional ML outcomes. Due to the fine-grained nature of learned features, DeepHACKS allowed deep phenotyping of subcellular dynamics, meaning that we can identify the rare subcellular phenotypes representing 1% of the entire time series data, which are specifically sensitive to molecular and environmental perturbations.

We applied DeepHACKS to cell edge dynamics, in which the leading edges of migrating epithelial cells undergo protrusion and retraction cycles under pharmacological and metabolic perturbations. Cell protrusion and retraction involves precise coordination of actin regulators to collectively organize actin

cytoskeleton[6-7,25-26]. Growing numbers of studies show the vital roles of cell protrusion, including in tissue regeneration[27-28] , cancer invasiveness and metastasis[29-31], and microenvironmental surveillance of leukocytes[32]. Dissecting such dynamics has been a challenging task due to substantial morphodynamic heterogeneity[6-7, 25] . DeepHACKS quantitatively identified the deep phenotypes of subcellular protrusion from highly heterogeneous and non-stationary-edge dynamics of migrating epithelial cells, revealing fine differential responses of leading-edge dynamics specific to perturbations. Particularly, DeepHACKS revealed the emergence of rare protrusion phenotypes upon blebbistatin or AICAR treatments. This suggests that the temporal features directly learned from leading-edge dynamics enable fine-grained identification of drug-related subcellular phenotypes, which may not be possible from static cell images. Our study opens up a new avenue for a comprehensive understanding of molecular mechanisms hidden in subcellular heterogeneity.

4.2 Materials and Methods

4.2.1 Cell Protrusion Dataset Collection

4.2.1.1 Experimental Description

The cell culture and live cell imaging procedures were followed according to the previous studies[6]. For the drug treatment experiments, we cultured PtK1 cells on 27mm glass bottom dishes (Thermo Scientific cat. #150682) for two days and stained with 5μgml^{-1} CellMask Deep Red (Invitrogen) following manufacturer's protocol. Then we monitored the cell using the microscopy. For Arp2/3 inhibition experiments, cells were incubated with 50nM of CK666 or CK689 (EMD Millipore) for an hour before imaging. For Cytochalasin D experiments, cells were incubated with DMSO or Cytochalasin D (Sigma) for half an hour before imaging. For Myosin inhibition experiments, cells were incubated with 20μM Blebbistatin (in short, Bleb) (EMD millipore, cat. # 023389) for half hour before imaging. For AMP-activated protein kinase (AMPK) inhibition experiments, currently, the cells were incubated with AICAR (1mM) (Sigma, cat. #A9978) for half hour before imaging. Besides, the cells were incubated with CC with low dose (2-10 μM) (EMD Millipore. cat. #171261) for half hour before imaging.

4.2.1.2 Dataset Description

In total, the whole dataset could be divided into 4 paired experiments: CK689/CK666, DMSO/CyD50/CyD100, DMSO/Bleb, Control/AICAR/CC. The details of the monitors cells and extracted

cell protrusions are described as Table 4.1. For ACF-based clustering and Deep Feature-based clustering method to pre-label the categories roughly, we did analysis for each paired experiment individually. The dataset of VASP and Arp2/3 are inherited from Chapter 3[7]. In total, we have 23802 samples. Besides, during the modeling training, we recruited another 9319 time series velocity protrusion samples not descripted here from our lab.

Regulator	Experiments	#video	#protrusions
Total	----	125	23802
VASP	DMSO	22	5961
	CyD50	16	3289
	CyD100	20	3199
Arp2/3	CK689	10	2340
	CK666	10	1425
Myosin	DMSO	14	2176
	Blebbistain	13	1920
AMPK	Control	6	1352
	AICAR	6	951
	CC	8	1189

Table 4.1 The dataset summary across different paired experiments.

4.2.1.3 Local sampling and event registration.

The procedures are the same as the previous study described in chapter 4.2.2.1. Here I will not describe them again. Mainly we first segment the cell edges and then cell edge velocity is calculated by tracking the cell edges using a mechanical model[25]. After that, along the cell boundary, the probing window with the size 500nm by 500 nm are generated to locally sample the velocity for further study.

4.2.2 ACF-based clustering

4.2.2.1 Calculation of partial dissimilarity matrix

In previous research[7], we demonstrated that ACF[35] is important to extract the distinguish protrusion phenotypes. However, ACF-based distance is only measured on the samples with equal length. Also, it's well known that there will be no auto-correlation for random noise signal. Therefore, in order to calculate the distance similarity between samples with similar length, we pad random noise to the shorter sample for further analysis. The details of procedure are listed as follows:

1) For each sample, we extracted the samples whose length is similar with the length of this sample below the threshold after the velocity samples are denoised by Empirical Mode Decomposition (EMD)[42]. Here, the default threshold is 6.
2) After that, among these samples, we pad random noise to make the length equal to the longest one. For each sample, we used the last five time points to estimate the mean value of random noise. Then, the missing part will be padded with random noise with estimated mean and standard variance.
3) Then, we represented the time series data by Symbolic ApproXimate representation (SAX)[33]. In SAX, we set the average interval (ratio) to four and use four symbolic to represent the data.
4) Finally, we calculate the Euclidean distance based on the auto-correlation coefficient.
5) Loop from step 1) to step 5) until all the samples are calculated.

6) At last, the final distance similarity matrix is the average mean from distance similarity matrix and its transpose to guarantee that the matrix is symmetric.

4.2.2.2 Clustering and Visualization

Community Clustering[34]: We applied community clustering method to partial similarity matrix. First, we made a K-nearest neighbor graph based on similarity distance. Here it's worth to mention the parameter *K* should be less than the minimum number of neighbors across the whole samples. Then, we calculate the adjacency matrix and detect the community using the R package *igraph*. Since we had no prior information about the optimal number of clusters in our variable-length protrusion data, the number of clusters was determined by our knowledge in Chapter 3 about the truncated protrusion phenotypes and testing across different experiments. Finally, we decided that the optimal number of clusters was 10.

Adjusted tSNE for visualization: If you input the raw data into t-SNE[36], the algorithm will first calculate sample-sample similarity based on the default Euclidean distance. For our special purpose, instead of inputting the raw data, we will replace the distance calculation step by loading the similarity matrix instead. Here, since we only calculate the similarity among samples with similar length, we will fill a larger value (we used 100 here) to the empty entry in our matrix.

4.2.3 Deep Features based clustering

4.2.3.1 Velocity time series preprocessing

In this step, we will non-linearly rescale the velocity to eliminate the effect of large magnitude of sample velocity since we assume that large magnitude with a high probability comes from less-accurate measurement and the majority of velocity magnitude should be less than 10μm/min based on our experience. Also based on our observation, we manually designed a mapping function using the sigmoid function implemented in Matlab.

$$vel\_recaled = 2 * (sigmf(vel\_raw, [0.3, 0]) - 0.5)$$

After scaling, the range of the velocity will become to [-1, 1], which fit well for further training in our deep learning framework.

4.2.3.2 training process in guided Bi-LSTM autoencoder

Dataset: Deep learning training always expect a large training dataset, usually larger than 10 thousand. To satisfy the requirement, we recruited more protrusion samples from other experiment in our lab. After that, the number of total samples is 27540. For training step, we randomly split the whole dataset into three parts: training set, validation set and test set with a ratio: 0.49, 0.21, 0.3. After that, we used the training set to fit the parameters of model and selected the best set of parameters in validation set.

Training process: The details of our proposed guided Bi-LSTM autoencoder[23, 37] is shown in Fig. 2.a. Mainly we use three layers of long short-term memory (LSTM)[38] as encoder to extract the features and use another three layers of LSTM as decoder to reconstruct the input. At the same time, in order to learn features related to our previous results from ACF-based clustering, we add a multilayer perception (MLP) classifier to guide the learned feature valuable to classify our previous label with a higher accuracy. The total loss included two parts: reconstruction loss using mean squared error function from autoencoder and classification loss using multiple-categorical cross-entropy function from MLP. Furthermore, since the loss value between cross-entropy and mean squared error is quite different, in order to balance the two parts fairly, we set ratio of loss weight between autoencoder and classifier to 25 to 1.

After that, we used training set to fit the parameters with the batch size 128 and 237 epochs on our GPU sever. During the training, we monitor the loss in the validation set to save the parameters for best performance. From the loss curve shown in Fig. 2.b, we observed that the model converges well. Then, we used the encoder parts to extract the features of the whole dataset for further analysis and predict the reconstructed dataset for comparison.

Besides the guidance Bi-LSTM autoencoder, we also test the performance of Bi-LSTM autoencoder and only MLP classifier to learn features using the same parameters and structure. Then we compared the difference by visualizing them using u-map, labelled the cluster label from ACF-based clustering.

4.2.3.3 Identification of phenotypes by clustering

After the features were extracted, we applied Principle Component Analysis (PCA) for feature reduction to eliminate the correlation between learned features. Based on the percentage of feature variance shown in Fig. 4.2.e, the first 15 components are saved for further analysis.

After feature reduction, the whole dataset was split into four paired experiments: CK689/CK666, DMSO/CyD50/CyD100, DMSO/AICAR, DMSO/Bleb. For each paired experiment, we calculate the sample similarity using Euclidean distance and then apply community detection to identify the distinct phenotypes.

To find the optimal number of clusters, we applied the external criteria: Davies-Bouldin Index (DBI)[39] and silhouette value[40] to estimate the optimal numbers in each experiment. We found that the optimal number of clusters is around ten, which is shown in Sup Fig. 1. Then, we applied community detection method for each experiment to obtain the phenotypes. After that, in order to make the number of cluster and the phenotype profiles consistent across all experiments, we decided that the optimal number of clusters was 11.

4.2.3.4 Visualization

For each paired experiment, we use t-SNE (t-distribution stochastic neighboring embedding)[36] for visualization with the default parameter (PC components: 15, perplexity: 30). However, for the comparison among guidance Bi-LSTM autoencoder, without guidance and without autoencoder, we utilize U-map[41] instead of t-SNE by considering the speed and time efficiency.

4.2.4 fine-grained phenotypes identification

After ACF-based and Deep feature-based clustering, we further sub-divided our interested clusters into fine-grained phenotypes using the ACF and Deep features separately. Here, differing from the previous clustering, we first pooled all the samples from the phenotypes of interest across different paired experiments together and then calculated the Euclidean distance for community clustering. After that, we split the dataset into different paired experiments for visualization and quantification. The optimal number of clusters was determined to achieve the maximum silhouette value. Specifically, we decided the optimal number of clusters for 'Accelerating' protrusion to be four while we decided six for the 'Bursting' protrusion.

4.2.5 Drug Perturbation Quantification

To evaluate the effect of our drugs like CK666 for Arp2/3 inhibitor. We first overlaid the velocity profiles between control and drug-treatment experiment together for each cluster or phenotype measure and then checked the effect of velocity magnitude visually. Then, we quantitatively measured the cluster proportion to represent the drug effect. We counted the number of each cluster in each cell (the number of windows) from both experiments. Then the distribution of the proportion were estimated using resampling strategy using *bootstrp()* in Matlab for 10000 times. After that, p-values were calculated by estimating the probability that proportion in one experiment is greater or less than that in another experiment. Also, the confidence intervals of each experiment were estimated by Matlab build-in function *bootci()*;

4.3 Results

4.3.1 DeepHACKS: Deconvolution of subcellular variable length protrusion heterogeneity.

Subcellular protrusion occurs over varying periods of times and creates a heterogeneous temporal length, which partly hinders the extraction of critical mechanistic details. To deconvolve the heterogeneity of the subcellular protrusion activity with variable length at fine spatiotemporal resolution, we developed a computational analysis pipeline, DeepHACKS (Fig. 4.1), which leverages the advantages of deep learning to extract the features automatically and prior information from our previous ML study[7]. We advanced our ML algorithms to include the short time samples (less than 250 seconds) with heterogeneous temporal lengths. DeepHACKS mainly contains three main compounds: i) ACF-based clustering (Fig. 4.1 c-e), ii) Deep Feature-based clustering (Fig. 4.1 f-g) and iii) Fine-grained phenotype identification and validation (Fig. 4.1 h-i). We used the ACF-based clustering to get the coarse cluster label as the prior information of our dataset. Then, Deep Feature-based clustering utilized the cluster label as a guidance to automatically learn features for further clustering. In addition to the advantage of

HACKS descripted in Chapter 4, DeepHACKS further allowed us to (1) systematically analyze the entire time series of protrusion velocities with variable temporal length to identify distinct subcellular protrusion phenotypes, (2) integrate prior ML information with deep feature learning, which makes outcomes consistent with the prior ML study, (3) utilize the learned features from deep learning to identify fine-grained protrusion phenotypes susceptible to molecular perturbations. The framework can provide a new avenue for better quantifying the effects of drug perturbation more precisely and comprehensively.

4.3.2 variable length time series clustering analysis of protrusion velocities.

As described in Chapter 3, we prepared our sample videos of PtK1 epithelial cells in control and drug-perturbation conditions (Fig. 4.1 a). After that, in each time-lapse move, we can observe the protrusion-retraction cycles in the leading edge. Then, we segmented the cell membrane boundary of each frame in each movie and locally divide the boundary into small windows with a size 500 by 500nm (Fig. 4.1 a). Then time series of protrusion velocities were acquired by averaging velocities of pixels in each window. After detecting the protrusion onset, the time-series protrusion velocities were aligned as a temporal fiduciary. Then, to eliminate the potential windowing drift and noise, we selected the samples, whose length is less than 50 frames (250 seconds) for further clustering analysis. After that, following the same procedure of HACKS in Chapter 3, we denoised the time series velocity profile using Empirical Mode Decomposition (EMD)[42] to trim the intrinsic fluctuation.

Since the sample length in our dataset is quite heterogeneous (Fig. 4.1 b), calculating the similarity between samples is a challenging problem. For the protrusion activities which are regulated by many different regulators, we assume that the time series protrusions are dissimilar if the length of them are too different. Therefore, instead of calculating the similarity distance among the whole samples, only the distance among samples with similar lengths are calculated.

In ACF-based clustering (Fig. 4.1 c-e), Auto-correlation functions as the time series features played an important role in discovering the novel phenotypes described in Chapter 3. However, ACF-based distance usually applies to measure the similarity among samples with equal length. Therefore, to calculate the ACF-based distance among time series with similar temporal lengths, we padded random noise to the later part of time series to make a uniform time length (Fig. 4.1 c). After we calculated the ACF-based similarity, we followed the HACKS pipeline including SAX (Symbolic Aggregate approximation)[33] and Euclidean-based ACF. We pooled all the distance similarity together for partial similarity distance (Fig. 4.1.d) and applied the community clustering algorithm[34], which has been shown to be robust and widely used in many applications, to determine the distinct clusters shown in Fig. 4.3-Fig. 4.6 for different paired drug-perturbation experiments.

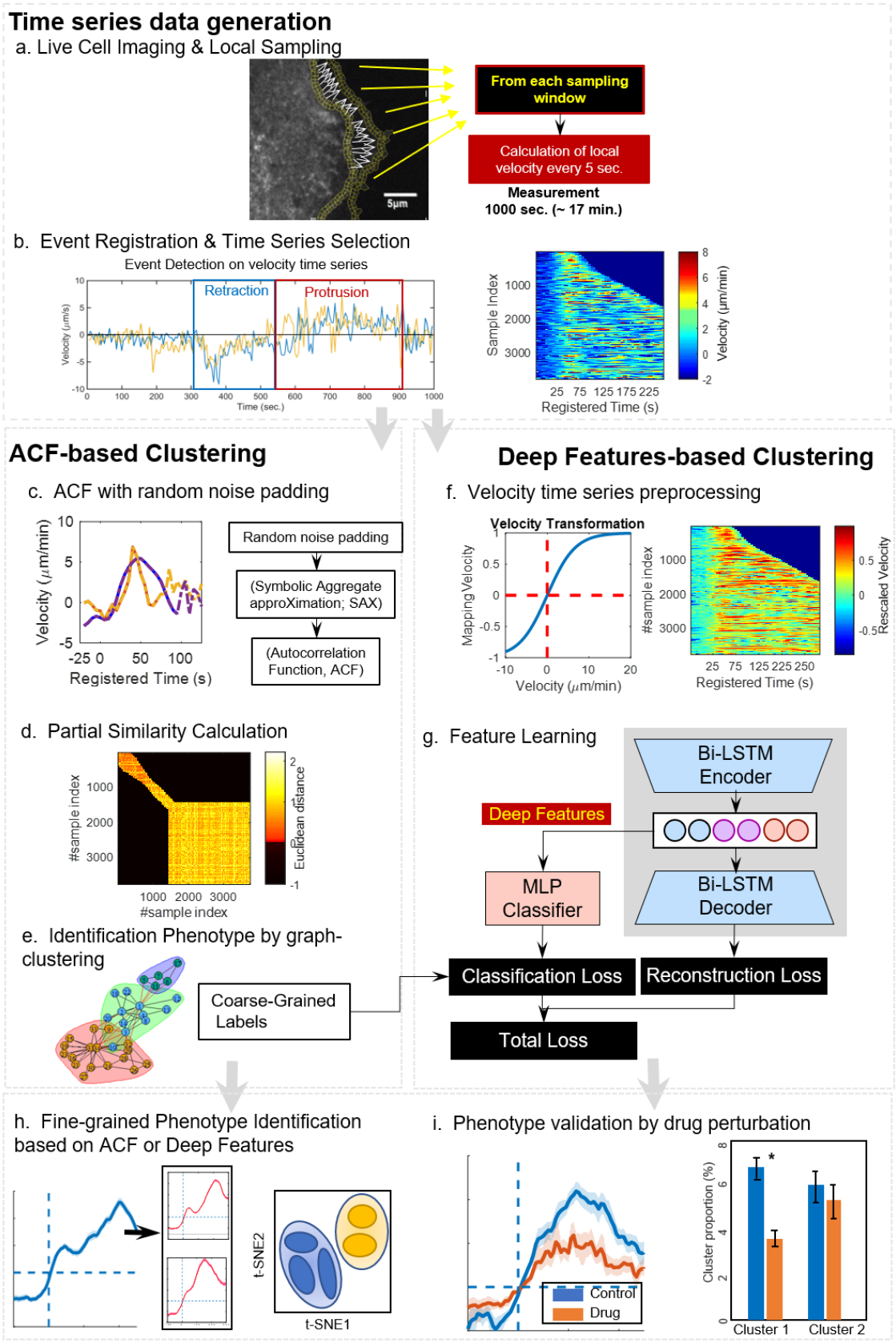


Figure 4.1 Schematic Representation of the Analytical Steps of DeepHACKS

In Deep Features-based clustering (Fig. 4.1 f-g), instead of manually designing features, we attempted to learn features automatically using the advantage of deep learning techniques. First, we rescaled the raw velocity to eliminate the effect of large velocity magnitudes and mapped them to the range [-1, 1] for further analysis (Fig. 4.1 f). Autoencoder[37] is widely used in many applications for feature learning by minimizing the difference between input and output as a reconstructed input. One weakness of basic autoencoder is that the features is difficult to interpret since there is no prior information included during the training process. Therefore, there are many structures[43-46] based on autoencoder are proposed like variational autoencoder[43]. Different from it by adding the prior information on the hidden layers, we added another classification branch to force the auto-encoder[46] to learn features which is efficient for category identification and consistent with our prior analysis (Fig. 4.1 g). In addition, we used Long-short term memory (LSTM)[38] , a deep learning structure specializing time series data. Particularly, LSTM can handle variable length time series by nature since it does not require a fixed length of time series. By Integrating these concepts, we proposed our feature learning framework called Guided Bi-LSTM autoencoder (Fig. 4.2 a). By optimizing the total loss together (Fig. 4.2 b-d), we trained Guided Bi-LSTM autoencoder using our dataset including more than 30 thousand samples. In Fig 4.2 f-g, by visually comparing the input with output, we can confirm that the reconstruction was performed well by our autoencoder. Afterwards, we applied Principal Compound Analysis (PCA) to reduce the feature dimension and selected the first 15 compounds whose explained variance is larger than 95% for further clustering (Fig. 4.2 e). Then we applied community detection to identify phenotypes. The results using Deep Features-based clustering will be discussed in Fig. 4.3-Fig. 4.6 in each experiment. Additionally, in order to validate the effectiveness of our Guided autoencoder, we trained another two models; i) the conventional autoencoder without guidance and ii) classifier training without autoencoder. Then we visualized our represented features labelled by the cluster results from ACF-based clustering using t-SNE techniques (Fig. 4.2 h-j). The visual comparison suggests that the features from Guided Bi-LSTM autoencoder are better organized on tSNE plots than two other methods.

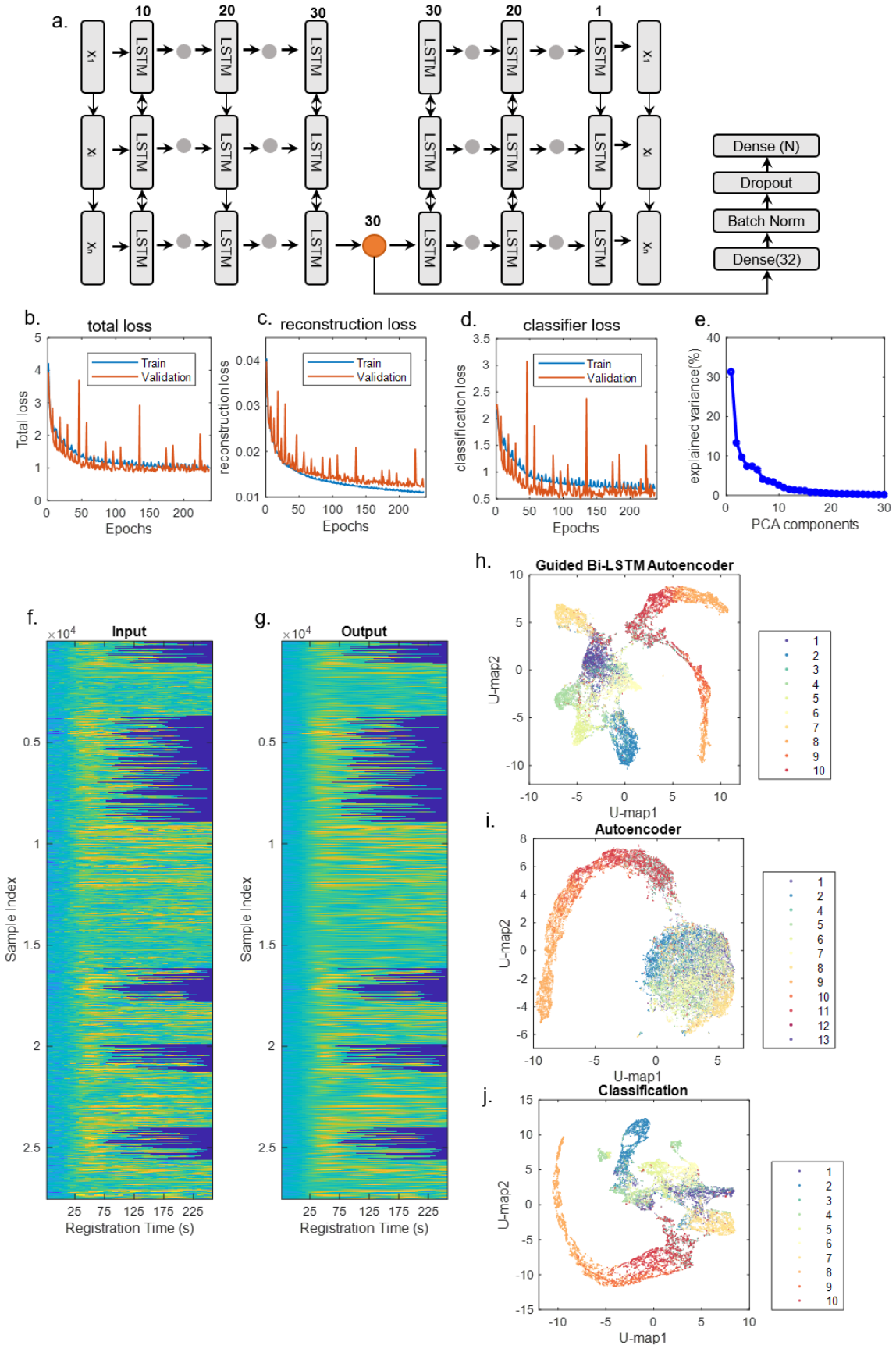


Fig 4.2 guidance Bi-LSTM autoencoder for Deep Features Extraction and Visualization

4.3.3 Identification of distinct subcellular protrusion phenotypes.

We applied ACF-based clustering and Deep Features-based clustering separately in four different drug perturbation experiments including CK689/CK666 (Fig. 4.3), DMSO/CyD50/CyD100 (Fig. 4.4), DMSO/Bleb (Fig. 4.5) and Control/AICAR/CC (Fig 4.6). Using the clustering criteria such as Davies-Bouldin Index[39], silhouette value[40], we evaluated the optimal number of clusters (Supplementary Fig.4.1a-b). Across the four experiments, we chose the optimal one as eleven, which were confirmed visually by the ordered distance maps and the silhouette plots (Supplementary Fig4.1 c-d) of the clustering results for Deep Features-based clustering while ACF-based clustering, we still chose at ten. Consistent with our previous discovery by HACKS, the visual inspection of the average velocity profiles (Fig. 4.3 a, Fig. 4.3.e, Fig. 4.4 a, Fig. 4.4 f, Fig. 4.5 a, Fig. 4.5.e, Fig. 4.6 a, Fig. 4.6 f) demonstrated that overall differences among the protrusion phenotypes in timing and number of oscillation periods. Across the four experiments, we were able to identify the similar clustering patterns using ACF-based clustering and Deep feature-based clustering, which suggests that our identified phenotypes are robust to sample selection and experimental variation. In addition to the previously identified phenotypes, including Cluster 1 named as 'fluctuation protrusion', Cluster 4/5/6 named as 'periodic protrusion' and Cluster 7 named as 'acceleration protrusion', we found another two novel phenotypes, Cluster 2/3 and three phenotypes from the samples with shorter temporal length (Cluster 8/9/10/11). Since Cluster 3 exhibited periodic edge velocity, we combined it with 'periodic protrusion'. Cluster 2 showed that edge velocity was changed dramatically with 100 seconds. Therefore, we named it as 'Bursting Protrusion'. For the convenience for the explanation, we still called them by cluster numbers hereafter.

Even though the average temporal patterns of protrusion velocities from ACF and Deep feature-based clustering are similar, the tSNE visualization revealed their significant differences (Fig. 4.3 b-c, Fig. 4.3 f-g, Fig. 4.4b-d, Fig. 4.4 g-I, Fig. 4.5 b-c, Fig. 4.5 f-g, Fig. 4.6b-d, Fig. 5.6 g-I). We can see that the tSNE plot of ACF-based clustering contains two large distinct clusters. The larger cluster mainly contains Cluster 1 to Cluster 7 while the smaller one includes Cluster 8 to 11. This comes from our strategy where we assume that time series samples with the different length belong to different clusters. However, Deep Feature-based clustering had much less distinct separation, which means the features learned automatically from Guided Bi-LSTM autoencoder are less affected by temporal length of velocity time series. Moreover, the large (Cluster 1-7) and small (Cluster 8-10) clusters in ACF-based clustering are tightly packed with sub-clusters while most of clusters from Deep Feature-based clustering are well separated. This suggests that Deep Feature-based clustering produce better clustering outcomes.

4.3.4 Deconvolution of heterogeneous drug responses in protrusion.

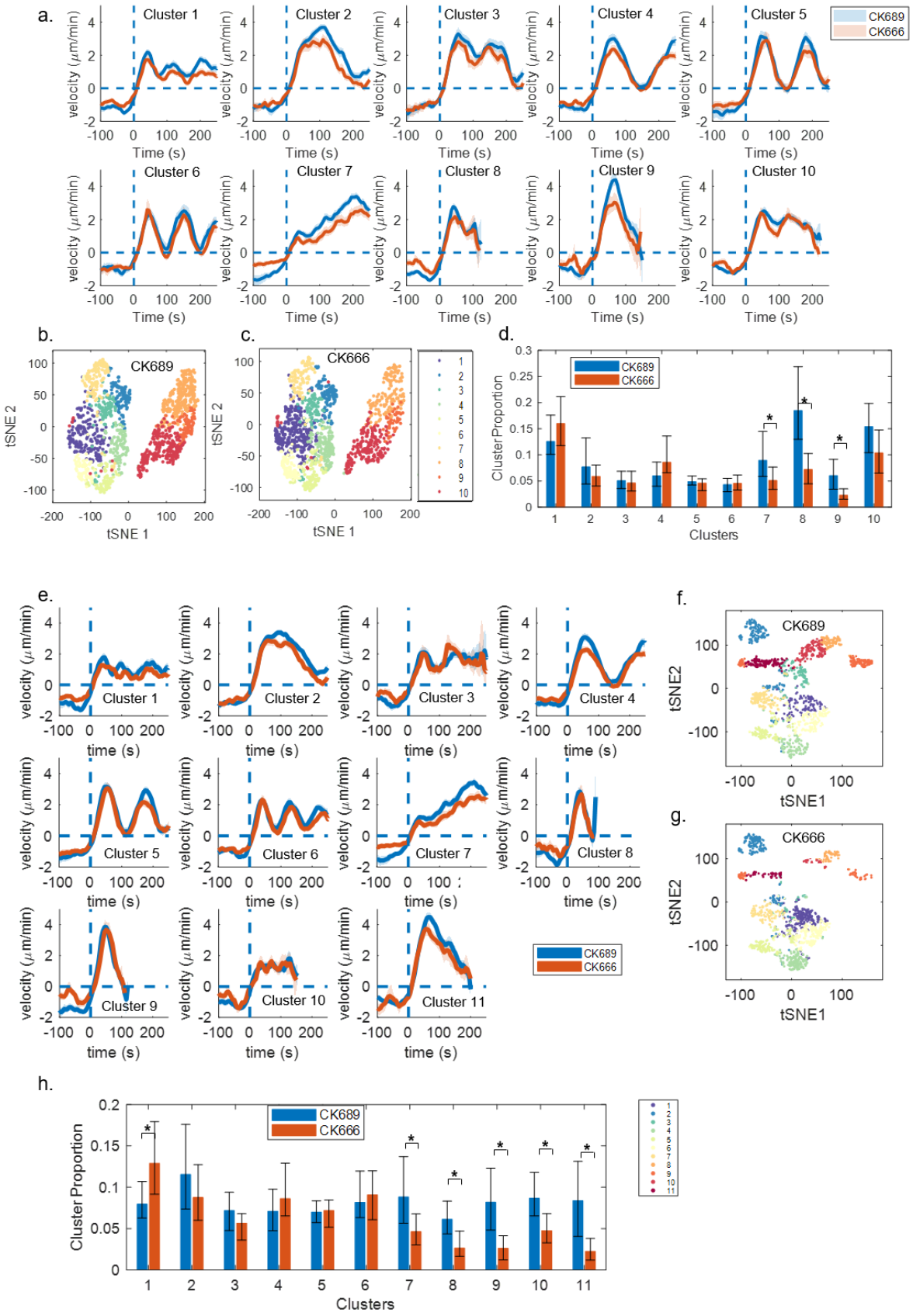


Fig 4.3 Subcellular Protrusion Phenotypes Revealed on paired Experiments CK689/CK666

To characterize the heterogeneous drug effects, we mainly focused on Cluster 7 (accelerating protrusion) and Cluster 2 (bursting protrusion). The p-values for each drug perturbation experiment are shown in Table 4.2 (CK689/CK666), Table 4.3 (DMSO/CyD), Table 4.4 (DMSO/Bleb) and Table 4.5 (Control/AICAR/CC). The statistical testing was performed with the proportions of each clusters if not mentioned specifically.

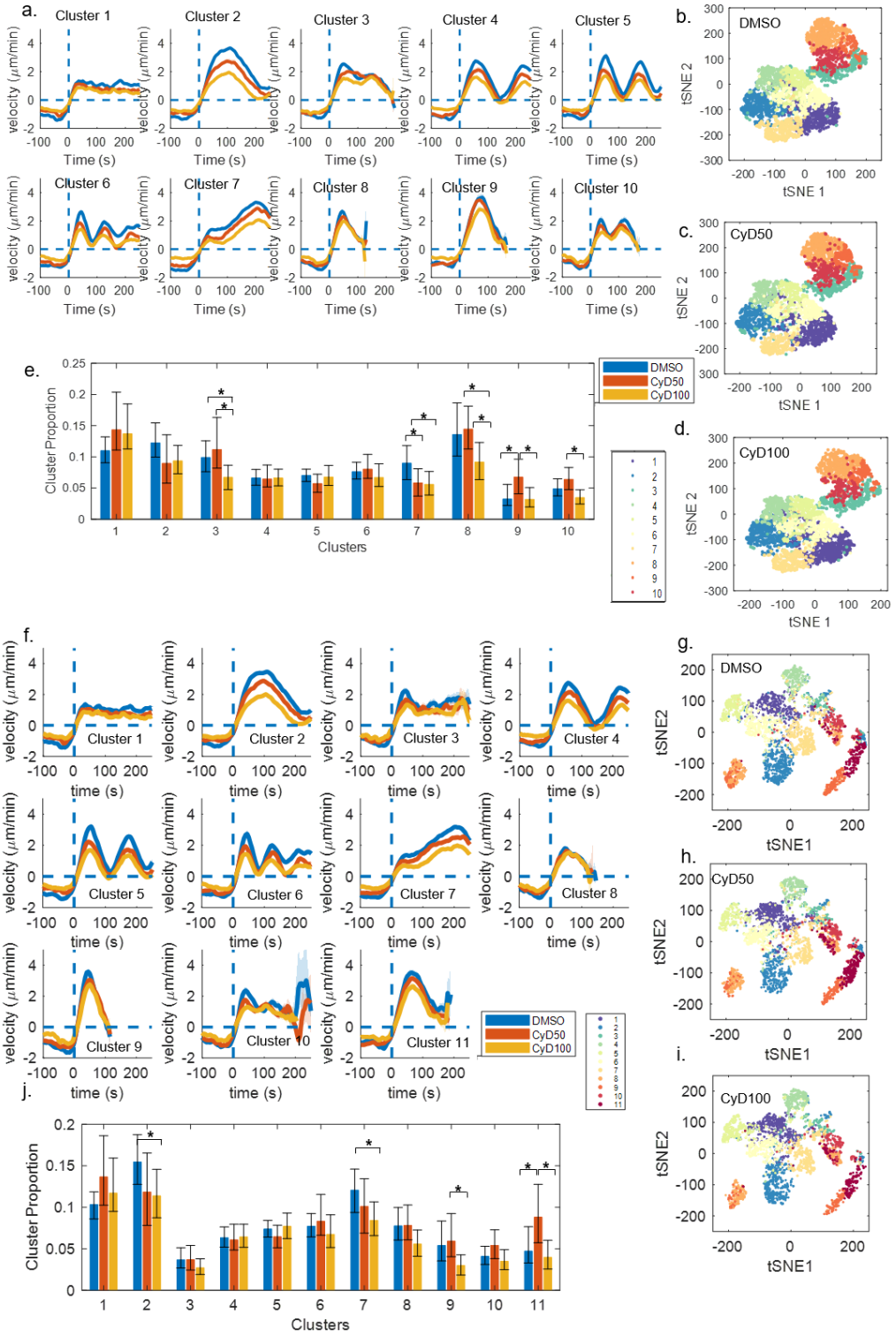


Fig 4.4 Subcellular Protrusion Phenotypes Revealed on paired Experiments DMSO/CyD50/CyD100

4.3.4.1 Cluster 7 “Accelerating protrusion” is consistent with previous analysis.

The drug effects of Cluster 7 by CK666 (Fig. 4.3 d, Fig. 4.3h), and CyD (Fig. 4.4e, Fig. 4.4j) are consistent with the results in Chapter 3. Both CK666 and CyD affected the protrusion proportion significantly (ACF-based clustering, CK666: p-value=0.0388; CyD50: p-value = 0.0361; CyD100: p-value = 0.0205 bootstrap sampling; Deep Feature-based clustering, CK666: p-value = 0.0151; CyD50: p-value = 0.1588; CyD100: p-value = 0.0107 bootstrap sampling;). The smaller p-values in Deep feature-based clustering further suggests the robustness of our phenotyping pipeline.

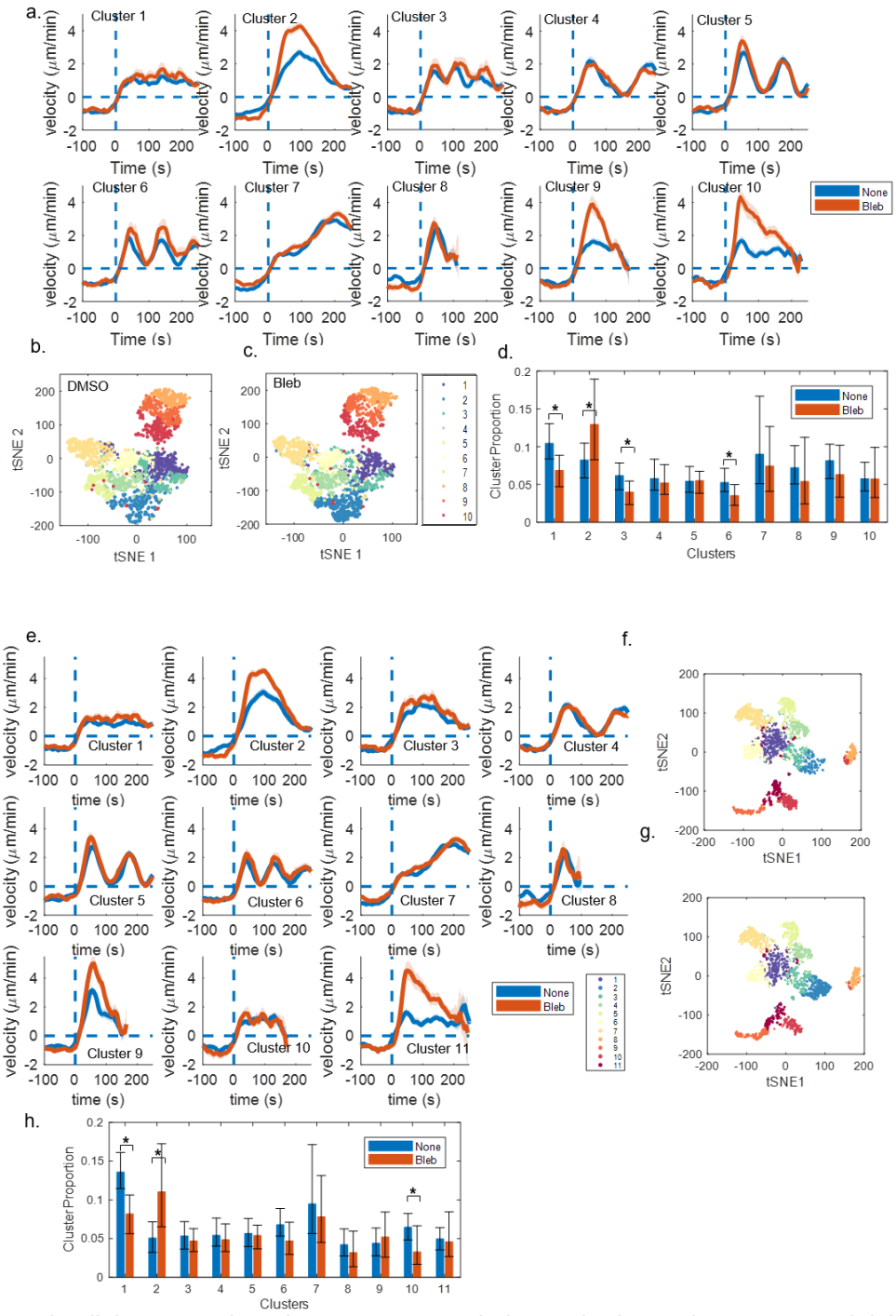


Fig 4.5 Subcellular Protrusion Phenotypes Revealed on paired experiments DMSO/Bleb

For the experiment treated with blebbistatin which inhibits the myosin II (Fig. 4.5d, Fig 4.5h), there is no significant effect (ACF-based features: p-value:0.3286 bootstrap sampling; Deep Feature-based clustering: p-value:0.2444 bootstrap sampling) in Cluster 7. However, in the experiments treated with AICAR or CC which activates or inhibits AMPK respectively, we found the significant effects in Cluster 7 in both ACF-based clustering (Fig. 4.6e) and Deep Features-based clustering (Fig. 6j). Specifically, for ACF-based clustering, we found the proportion in AICAR experiment was significantly larger than that in CC experiment (p-value: 0.0076 bootstrap sampling) while the significance between in AICAR and Control experiments is not detected (p-value: 0.0685 bootstrap sampling). However, for Deep Features-based clustering, the significance between AICAR and control experiments is detected successfully (p-value: 0.0348 bootstrap sampling). This suggests that AMPK increases the proportion of Cluster 7. The results also suggest Deep Feature-based clustering was more sensitive to detect subtle perturbation effect than ACF-based clustering. This result suggests that in order to generate the velocity profile in Cluster 7 ('accelerating' protrusion), in addition to the role of Arp2/3 branching action networks and VASP elongating actin filaments, energy metabolism is also required. Since the velocity in Cluster 7 keeps increasing until 200 second after protrusion onset, it is plausible that the leading edges need more force and energy to push the membrane forward and overcome increasing membrane tension or other retracted force.

4.3.4.2 Cluster 2 "Bursting Protrusion" is affected both by Cytochalasin D and Blebbistatin, but not affected by CK666.

Next, we investigate the effects of CK666, Cytochalasin D, and blebbistatin in Cluster 2 (Bursting Protrusion) using ACF-based and Deep Features-based clustering. In contrast to Cluster 7, CK666 (Fig. 4.3d, Fig. 4.3h) did not show significant effects (ACF-based clustering: CK666 p-value: 0.2247; Deep Features based clustering: CK666 p-value: 0.1622; bootstrap sampling). In the experiment treated with CyD to perturb VASP (Fig. 4.4e, Fig. 4.4j), we found 100nM CyD significantly affected the Cluster 2 while the effect of 50nM CyD was not detected. (ACF-based clustering: CyD50 p-value: 0.0831; CyD100 p-value: 0.0518; Deep Feature-based clustering: CyD50 p-value: 0.0947; CyD100 p-value: 0.0297 bootstrap sampling).

Furthermore, in both the Bleb-treated and AICAR-treated experiment, the velocity magnitude was dramatically increased (Fig. 4.5a, Fig. 4.5e, Fig. 4.6a, Fig.4.6f). The quantification of the proportion of Cluster 2 confirmed that Cluster 2 was significantly increased by the Bleb treatment in both ACF-based clustering (Fig. 4.5d, p-value: 0.0499 bootstrap sampling) and Deep Features-based clustering (Fig. 4.5h, p-value: 0.0137). Moreover, when cells were treated with AICAR, both Deep Feature-based clustering and ACF-based clustering did not detect the difference (Fig.4.6e ACF based clustering: p-value:0.075 Deep Features based: p-value: 0.0555 bootstrap sampling). Both ACF-based clustering and Deep

Features-based clustering suggested no difference between AICAR-treated and control experiment (Fig. 4.6e: p-value 0.1381, Fig. 4.6j p-value: 0.0949 bootstrap sampling).

For 'Bursting' protrusion, the velocity was accelerated dramatically in a very short time interval (50 second) at protrusion onsets, where the retraction force generated by Myosin II plays a positive role to pull the membrane back. The retraction force will hinder the acceleration at the early stage. If the function of Myosin is inhibited, it will be more efficient to generate more forward motion since less retracted force is generated at the protrusion onsets. Therefore, the results suggest that for 'Bursting' protrusion with large acceleration at the early stage, is more influenced by Myosin II hindering the leading edges to move forward as a negative effect. Moreover, even though in AICAR experiment, the cell proportion is not detected significantly, we still could observe that the velocity magnitude is increased clearly. It suggests that energy metabolism still affect the 'Bursting' protrusion.

4.3.4.3 Protrusion phenotypes with short temporal length are affected by CK666, Cytochalasin D and AICAR

In addition to Cluster 7 and Cluster 2, we further investigate the drug effects on three more phenotypes (Cluster 8-11) generated from time series with shorter temporal lengths. In CK666-treated experiment, the proportion of Cluster 8 was significantly decreased (Fig. 4.3d ACF-based clustering p-value: 0.00029; Deep Features-based clustering p-value: 0.0022 bootstrap sampling) and the proportion of Cluster 9 was also significantly affected by CK666 (Fig. 4.3h ACF-based clustering p-value: 0.0048 Deep Feature-based clustering p-value: 0.0001 bootstrap sampling). The significant decreased proportion of Cluster 10 was detected by Deep Features-based clustering (Fig. 4.3h p-value: 0.0062 bootstrap sampling) while not detected by ACF-based clustering (Fig. 4.3.d p-value: 0.0576 bootstrap sampling). Furthermore, the proportion of Cluster 11 was also significantly decreased by Deep Feature-based clustering (Fig 4.3h p-value: 0.0028, bootstrap sampling). It suggests that CK666 affects all short protrusion phenotypes significantly. In CyD-treated experiments, these four phenotypes were sensitive to the dose of the CyD. There was no clear effect on all the clusters with a higher dose (100nM) while CyD with a lower dose (50nM) significantly increased the proportion of Cluster 11 (Fig. 4.4.d p-value: 0.006 bootstrap sampling). The details of testing results were shown in the Table 4.3. The results suggest that even though Arp2/3 only affect some long protrusion phenotypes such as "Acceleration" protrusion, however, Arp2/3 play an important role for all phenotypes of the short protrusion events, which is consistent with the well-accepted migration model that Arp2/3, branching the action network, is essential to cell migration. Also, It's worthing to mention that for the long protrusion event, Arp2/3 only affect 'Acceleration' Protrusion and some periodic protrusion phenotypes.

In Bleb-treated experiment, the comparison of velocity profile suggests that Cluster 9/10 identified by ACF-based clustering were significantly increased by blebbistatin, while Cluster 9/11 identified by Deep

Features-based clustering were significantly affected by blebbistatin. The proportion of Cluster 8/9/11 had no effect while the significantly decreased proportion of Cluster 10 was detected by Deep Features-based clustering (Fig4.5h p-value: 0.0177 bootstrap sampling).

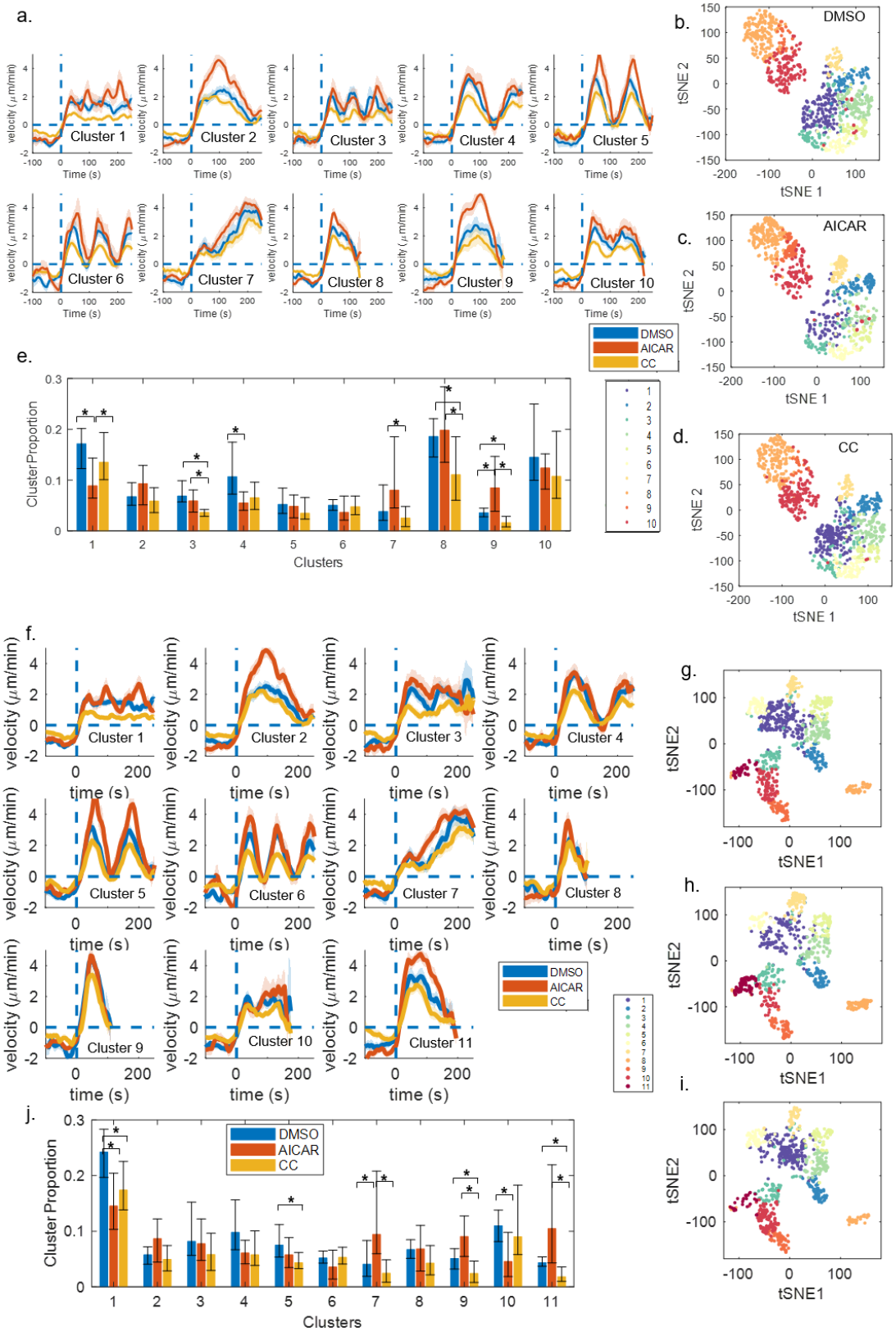


Fig. 4.6 Subcellular Protrusion Phenotypes Revealed on paired experiments Control/AICAR/CC

In AICAR-treated experiment, both clustering methods detected different phenotypes. In ACF-based clustering, both velocity profile (Fig.4.6a) and proportion test (Fig.4.6e, control vs AICAR p-value: 0.0278; bootstrap sampling) suggest that Cluster 9 was significantly increased by AICAR. The proportion test (Fig. 4.6e, cluster 8: control vs CC p-value: 0.0297, cluster 9: control vs CC p-value: 0.0040, bootstrap sampling) suggests that Cluster 8/9 were significantly decreased by CC. While in Deep Features-based clustering. The velocity profile (Fig. 4.6f) suggests that Cluster 11 was affected by AICAR while the proportion test was not detected significantly. The proportion in Cluster 10 was only significantly decreased by AICAR (p-value: 0.0037 bootstrap sampling). Furthermore, the proportion of Cluster 9 and Cluster 11 were decreased by CC. (Cluster 9, p-value: 0.0247; Cluster 11, p-value:0.0007 bootstrap sampling). The details of testing results were shown in the Table 4.5. Also, it is important to note that ACF-based clustering and Deep Features-based clustering identified different phenotypes for AICAR/CC treatment dataset. The results suggest that for energy metabolism play some role in the shorter protrusion since CC deactivated AMPK to generate less energy will reduce the proportion of Cluster 9/11.

Taken together, using our pipeline we were able to identify novel subcellular protrusion phenotypes with short temporal lengths. Furthermore, in drug-treatment experiments we confirmed that these phenotypes were differentially affected by various drug perturbations and CK666 significantly affected all short protrusion phenotypes.

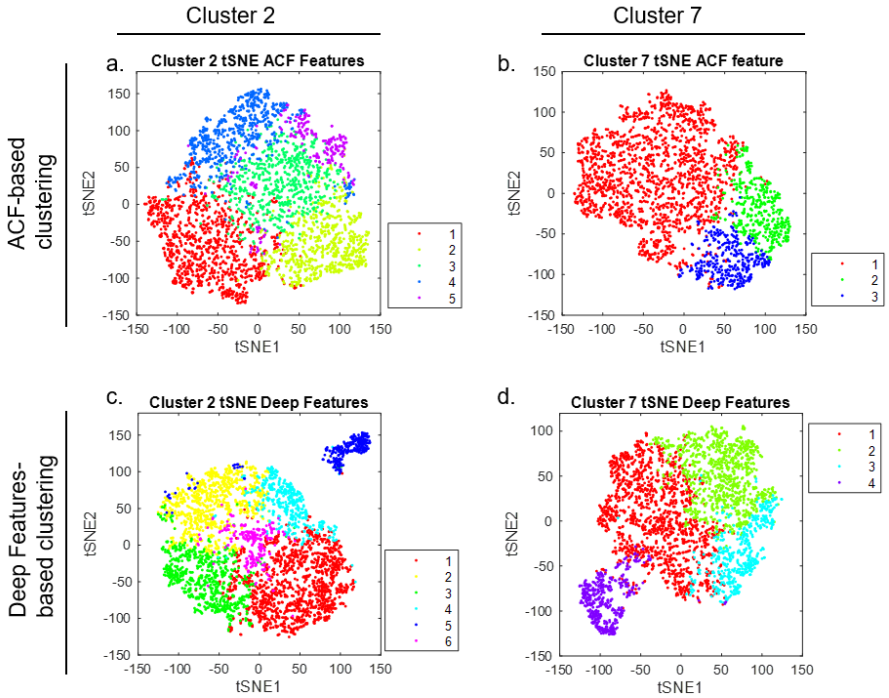


Fig. 4.7 t-SNE on Cluster 2 and Cluster 7 from ACF-based Clustering and Deep Features-based Clustering

4.3.4.4 Fine-grained phenotypes identification in Cluster 2 and Cluster 7.

Previously, we found Cluster 2 ('Bursting protrusion') and Cluster 7 ('Acceleration protrusion') had quite different drug characterizations. Therefore, we isolated the samples from Cluster 2 and Cluster 7 and

visualized the data distribution using t-SNE by ACF and Deep Features respectively in (Fig. 4). Notably, t-SNE plot by Deep Features (Fig. 4.7.c, d) suggested that the samples in Cluster 2 and Cluster 7 should be further divided, which may have differential susceptibilities to different drugs. We called this process fine-grained phenotypes identification since the previous identified phenotypes would be sub-divided to further investigate the effects of different drugs for better fine-grained phenotypes identification. Here, the number of clusters was determined by the condition that the samples in the isolated group in t-SNE visualization was isolated as an individual group. After that, the number of clusters in ACF-based clustering was adjusted accordingly.

The fine-grained phenotypes identified in Cluster 7 were shown in Fig. 4.8 (ACF-based clustering) and Fig. 4.9 (Deep Features-based clustering). In ACF-based clustering, we could identify three clusters with different profiles while in Deep Features-based clustering, we found one more cluster (Cluster 4 located at the right in Fig. 4.9.a-e), which was not significantly affected by any drugs (p-value are saved in Table 4.7). Compared with ACF-based clustering, Deep Features-based clustering generally identified clearer patterns and the drug characteristic, which was consistent with our previous study in Chapter 3. In details, we identified weak cluster (sub-cluster 1) and two strong clusters (sub-cluster 2 and sub-cluster 3). The weak cluster was not affected using proportion test by CK666 (Table 4.7, p-value: 0.0846, bootstrap sampling), AICAR (Table 4.7, p-value: 0.2602, bootstrap sampling) and CC (Table 4.7, p-value:0.4664 bootstrap sampling) while it was affected by low dose of CyD (Table 4.7, p-value: 0.0353, bootstrap sampling). However, strong clusters were generally affected by all the drugs (the details results are shown in Table 4.7). For Bleb-treatment experiment, both ACF-based clustering and Deep Features-based clustering did not find any significant effect on all fine-grained phenotypes (Table 4.7). The sub-phenotypes results suggest that Deep features-based cluster can better recognize the fine-grained phenotypes.

In Cluster 2 (‘Bursting protrusion’), Deep Features-based clustering identified 6 clusters (Fig. 4.11) and ACF-based clustering identified 5 clusters (Fig. 4.10). For Bleb-treatment paired experiment, we could find that sub-cluster 1 was significantly affected by the proportion test on both ACF-based clustering (Table 4.8, p-value: 0.0090, bootstrap sampling) and Deep Features-based clustering (Table 4.8, p-value: 0.0003, bootstrap sampling) while others sub-clusters had no significant effect by Blebbistatin perturbation using the proportion test. By comparing the proportion test without fine-grained phenotypes identification, the p-value in ACF-based clustering (Table 4.4, p-value: 0.049, bootstrap sampling) and Deep Features-based clustering (Table 4.4, p-value: 0.013, bootstrap sampling), is much larger than those after fine-grained phenotypes. Moreover, Cluster 1 was also significantly affected by AICAR (p-value: 0.0233, bootstrap sampling), which was not detected on the ‘Bursting protrusion’. Furthermore, velocity profile also suggested that Cluster 1 identified by ACF-based clustering and Deep Features-based clustering were both significantly affected by blebbistatin and AICAR. These results suggest that

our fine-grained phenotypes identification step could help to isolate the phenotypes for better quantitation of sensitivity of drugs.

Taken together, using our fine-grained phenotype identification, we were able to refine our discovered protrusion phenotypes by eliminating unrelated samples for better characterization of drug effects. Previously, the 'Acceleration' and 'Bursting' protrusions were defined by the velocity profile. After the fine-grained phenotypes, we could redefine the 'Acceleration' and 'Bursting' protrusions based on the regulator dynamics. Mainly, 'Acceleration' protrusion (sub-cluster 2/3) is affected by the drug CK666, CyD and AICAR while 'Bursting' (sub-cluster 1) protrusion is regulated by the Myosin II and AMPK. By comparing the fine-grained phenotypes from ACF-based and Deep Features-based clustering, we found that fine-grained phenotypes from Deep Features-based clustering provide better quantification for statistical analyses.

4.4 Discussion

Not only temporal profile patterns of subcellular protrusion, but also the temporal length is highly heterogeneous. By advancing HACKS described in Chapter 3, which dissects the heterogeneity of temporal profile patterns, we have further demonstrated that our new computational framework DeepHACKS could effectively deconvolve both two aspects of heterogeneity in temporal profile patterns and temporal length simultaneously and revealed specific phenotypes susceptible to pharmacological perturbations. To our best knowledge, our study is the first to propose a pipeline to handle the temporal length heterogeneity in subcellular protrusion activities for the identification of hidden phenotypes, although some previous studies have generated and examined the temporal dynamics of cell protrusion at the cellular level[47-49]. Using the drug-perturbation experiments, we refined the characterization of 'acceleration protrusion' susceptible to the pharmacological perturbations. Further, we identified another novel phenotype called 'Bursting protrusion', which was specifically susceptible to myosin II inhibitor, blebbistatin. Although previous studies have claimed the role of mosyin II in cell protrusion [50-51], we first dissected this protrusion phenotype at a fine spatiotemporal scale and further dissect the 'Bursting protrusion' by fine-grained phenotyping to better characterize the heterogeneity of drug effects.

Deep learning achieved many successes in biological and biomedical image areas. However, the application to biological time series has been limited. Here, we collected more than 30,000 protrusion velocity time series and proposed a Bi-LSTM autoencoder framework to automatically extract temporal features, which is particularly helpful for temporal phenotyping. Furthermore, we were able to integrate the prior ML results with Bi-LSTM autoencoder to learn important features consistent with the prior analysis. From our results in t-SNE visualization and drug sensitivity analysis, we found that features learned automatically from our framework could capture better characteristics of dataset and have better sensitivity to detect drug effects. Although in deep learning or machine learning fields, there are several

strategies proposed to integrate prior information[43-46], we are the first to propose an effective deep learning framework with prior information for unsupervised learning of time series data. Moreover, our framework is not restricted to use cluster label information. Potentially, we could integrate deep learning with many different types of prior information, like types and doses of different drugs. Also, our framework could directly extend to multivariate time series data without modification. For example, we could integrate the fluorescence information, position information and even acceleration with the velocity profiles as multivariate time series data to fit our model to identify the phenotypes.

Moreover, for unsupervised learning methods, it is always an issue to determine the optimal number of clusters. There are many different strategies like silhouette value or DBI proposed to address this issue. However, these strategies are developed on the general characteristics of clustering such as cohesion in the same clusters and separation between different clusters, which is independent with any specific problems. Therefore, for the specific biological questions, it is more reasonable to utilize them to estimate the range of optimal numbers of clusters instead of the optimal number. Then, the optimal number of clusters should be determined based on the domain knowledge. Therefore, we suggest that for unsupervised learning application in cell biology, the optimal number of clusters should be determined by the integration of different computational strategies and specific domain knowledge.

Furthermore, in current stage, our pipeline was only evaluated to detected protrusion samples in PtK1 cell line. Based on the previous study of MCF10A in Chapter 3, we expect that our discovered phenotypes with regulator dynamics could be identified consistently in other epithelial cell lines like MCF10A. Furthermore, our pipeline, extracting features by integrating the advantage of deep learning and domain knowledge could directly extend to detect the rare population in other biological research such as small populations with drug sensitivity in drug discovery.

Here, we demonstrate the power of machine learning or deep learning to handle heterogeneous datasets, which are ubiquitous in biomedical and biological fields[52]. Also, even if prior information is always valuable in biomedical and biological fields, it is a challenge how to incorporate prior information to deep learning framework. We expect our deep learning framework, DeepHACKS incorporating prior information into autoencoder will accelerate understanding of mechanism of heterogeneous cellular or subcellular activities.

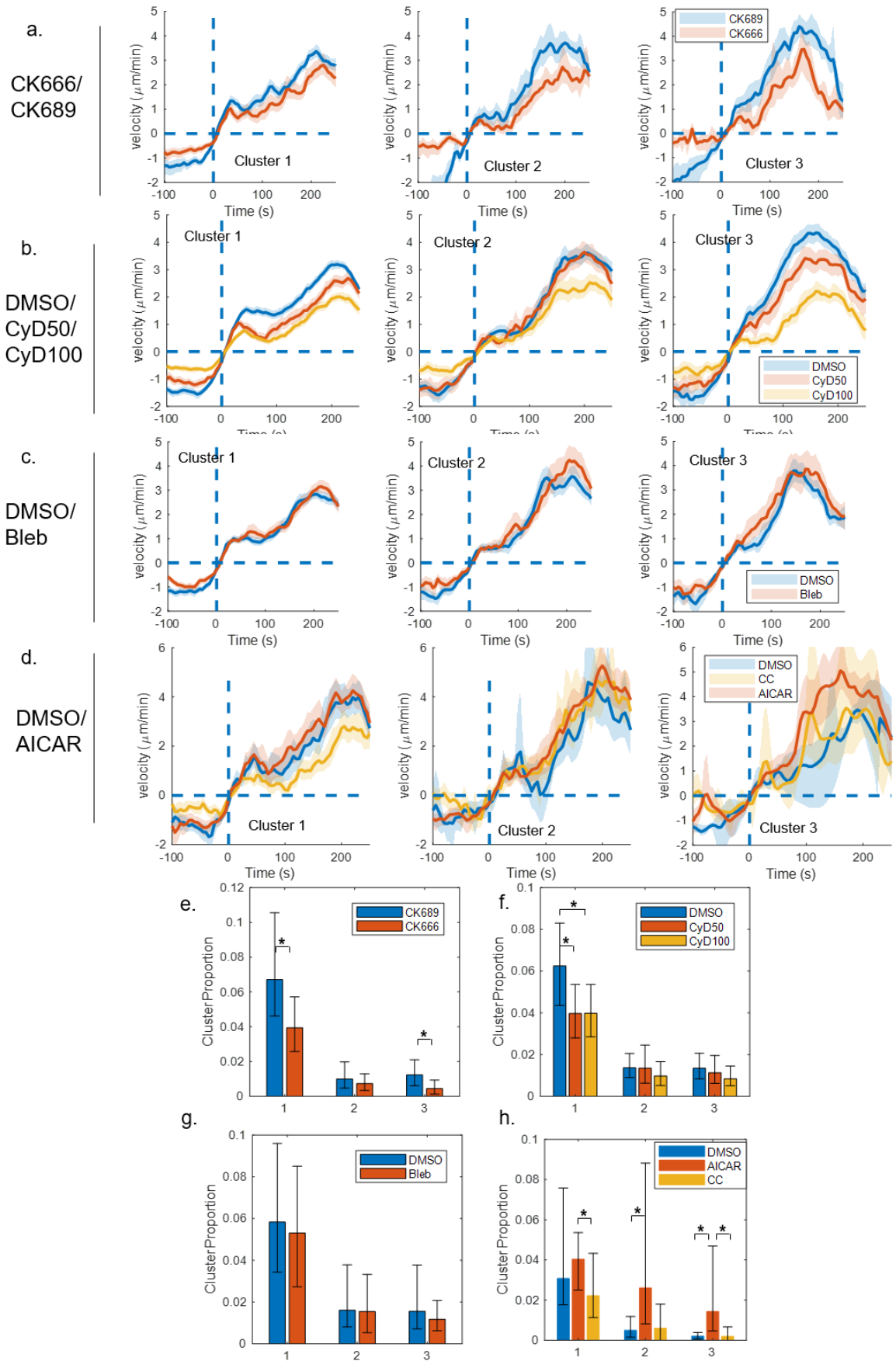


Fig. 4.8 Fine-grained Phenotypes in Acceleration Protrusion (Cluster 7) revealed by ACF-based Clustering

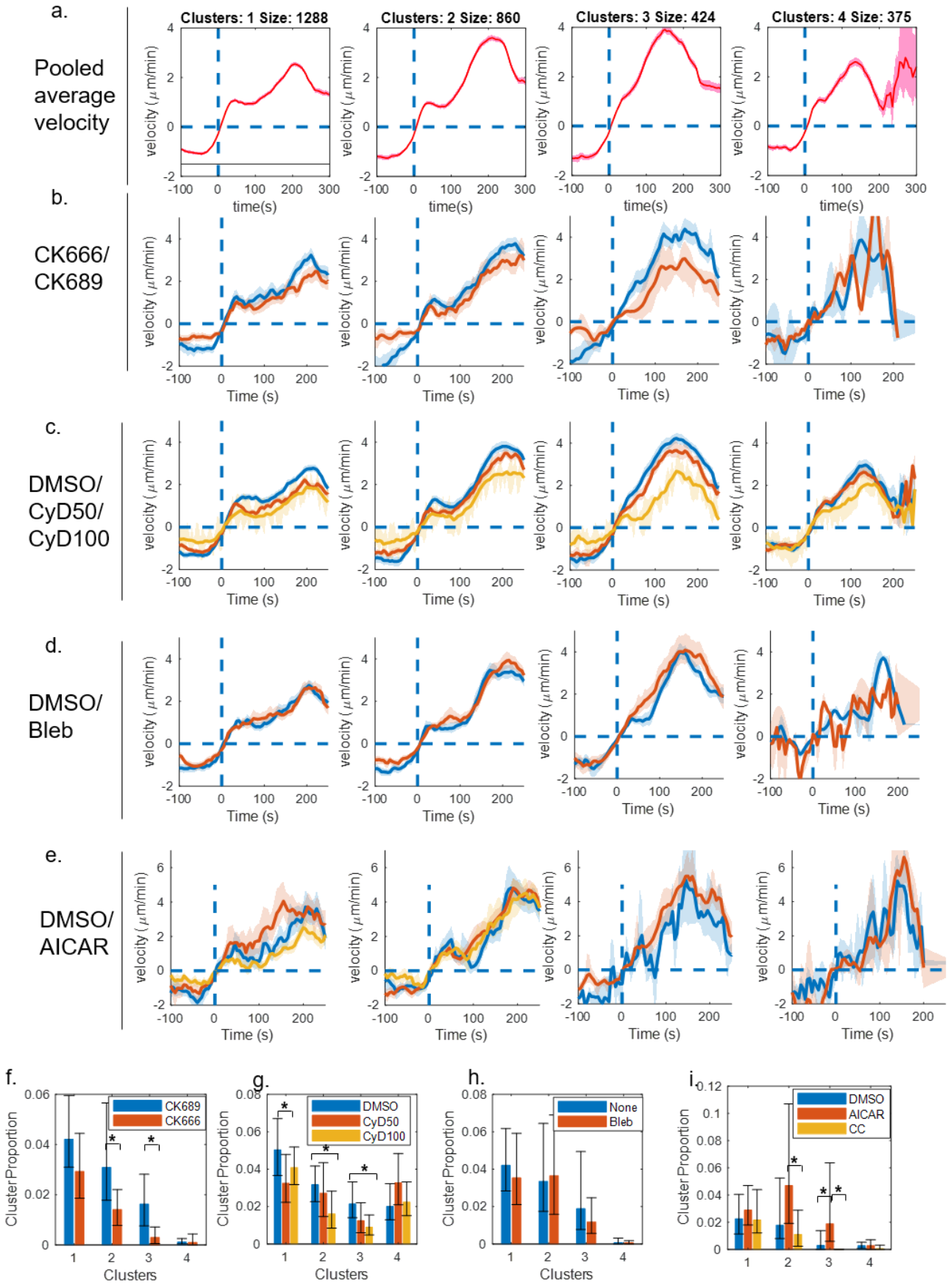


Fig. 4.9 Fine-grained Phenotypes in Acceleration Protrusion (Cluster 7) by Dee Features-based Clustering

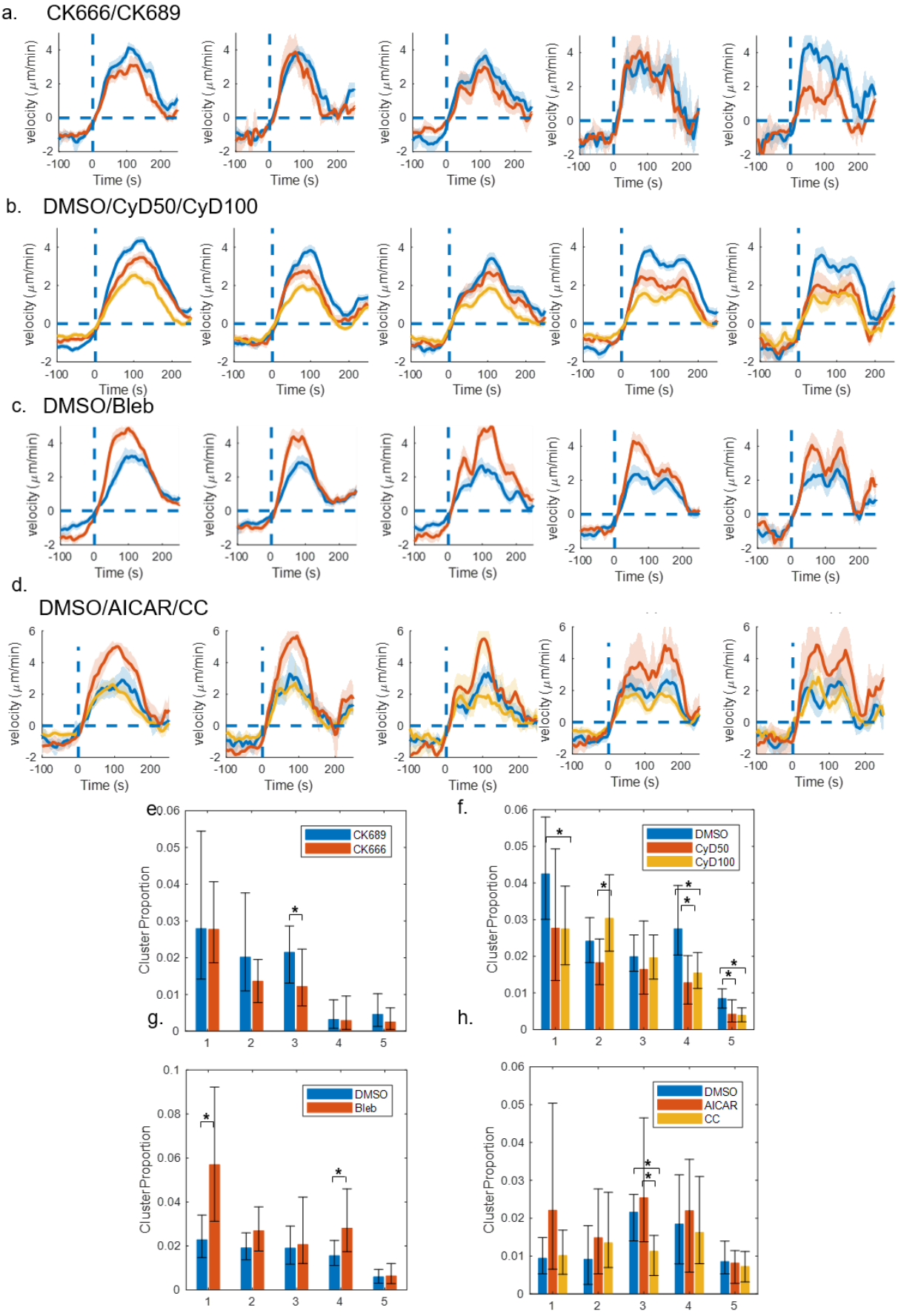


Fig. 4.10 Fine-grained Protrusion Phenotypes in Bursting Protrusion (Cluster 2) by ACF-based Clustering

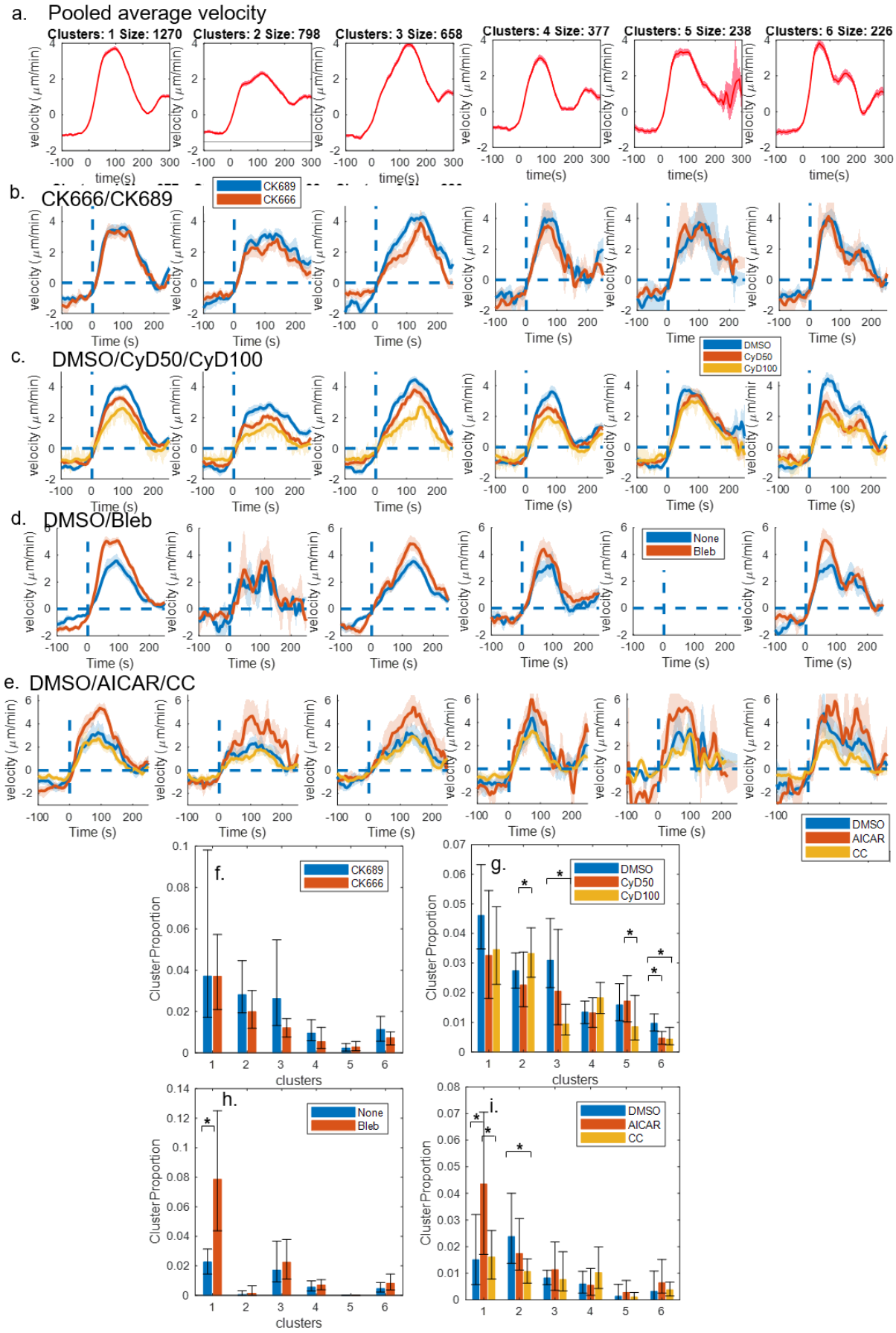
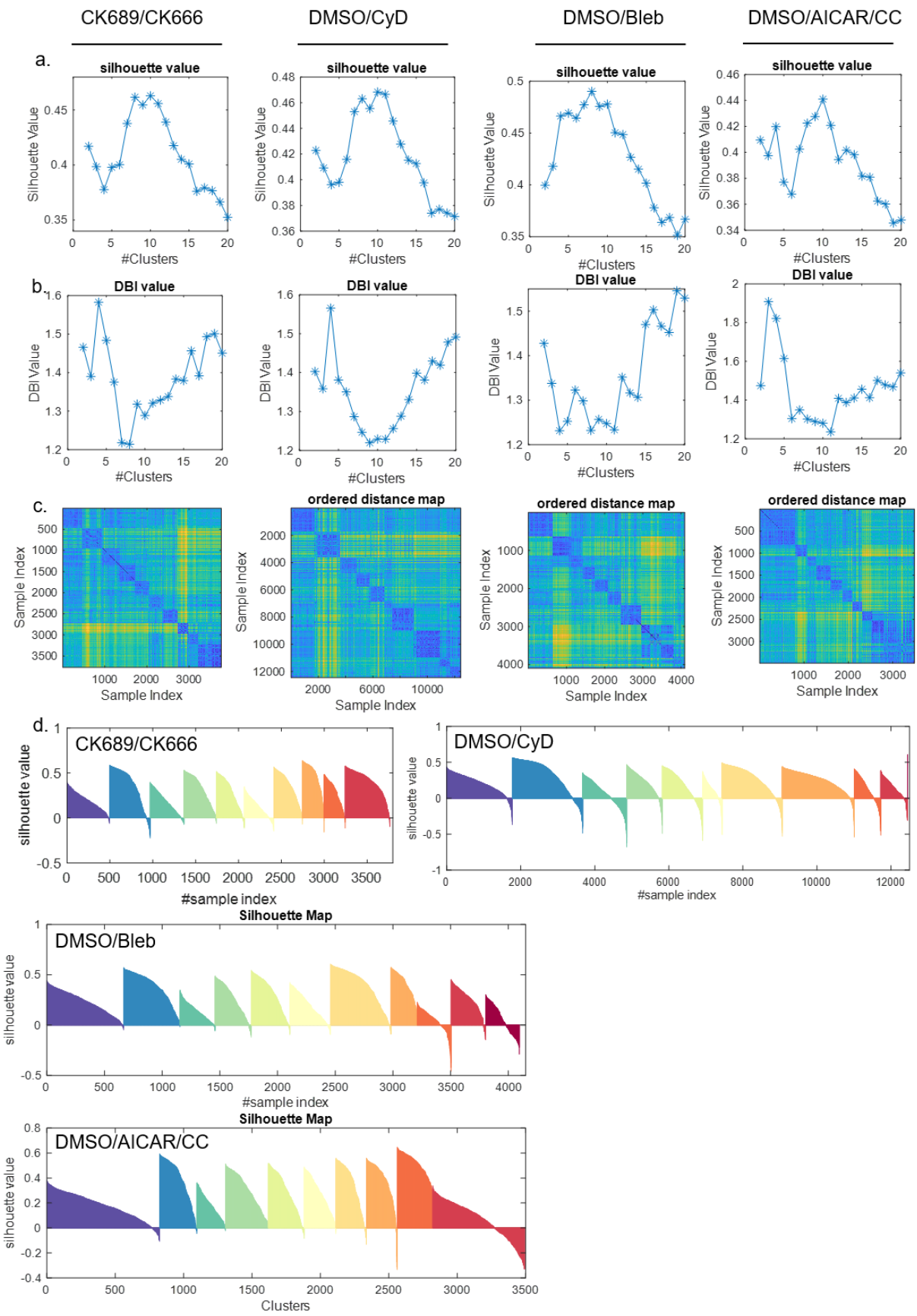


Fig. 4.11 Fine-grained Phenotypes in Bursting Protrusion (Cluster 2) revealed by Dee Features-based Clustering



Supplementary Fig. 1: The Optimal Number of Clusters Evaluation.

Table 4.2: The p-values for statistical analysis of cell proportion in CK689/CK666

Cluster index	1	2	3	4	5	6	7	8	9	10	11
P-value(ACF-based)	0.1251	0.2247	0.357	0.0752	0.356	0.397	0.0388	0.00029	0.0048	0.0576	-
P-value (Deep Features based	0.0177	0.1763	0.1432	0.2158	0.414	0.3253	0.0190	0.0022	0.0010	0.0062	0.0028

Table 4.3: The p-values for statistical analysis of cell proportion in DMSO/CyD

Cluster index		1	2	3	4	5	6	7	8	9	10	11
P-value (ACF-based)	DMSO vs CyD50	0.0773	0.0831	0.2977	0.4353	0.0644	0.3690	0.0361	0.3699	0.0187	0.0920	-
	DMSO cv CyD100	0.0756	0.0518	0.0218	0.4952	0.3854	0.2072	0.0205	0.0440	0.4968	0.0548	-
	CyD50 vs CyD100	0.4128	0.4128	0.0137	0.4419	0.1647	0.1526	0.4365	0.0127	0.0159	0.0037	-
P-value (Deep Features based	DMSO vs CyD50	0.0584	0.0947	0.4988	0.3918	0.1374	0.3438	0.1702	0.4827	0.3820	0.1062	0.023
	DMSO cv CyD100	0.2226	0.0297	0.1049	0.4592	0.3735	0.2046	0.0137	0.0392	0.0301	0.2404	0.298
	CyD50 vs CyD100	0.2210	0.4394	0.1353	0.3559	0.1194	0.1453	0.1952	0.0434	0.0129	0.0396	0.006

Table 4.4: The p-values for statistical analysis of cell proportion in DMSO/Bleb

Cluster index	1	2	3	4	5	6	7	8	9	10	11
P-value(ACF-based)	0.008	0.049	0.035	0.338	0.459	0.0419	0.328	0.2309	0.1894	0.493	--
P-value (Deep Features based	0.0007	0.013	0.287	0.3405	0.4368	0.0594	03189	0.2444	0.324	0.017	0.387

Table 4.5: The p-values for statistical analysis of cell proportion in DMSO/AICAR/CC

Cluster index		1	2	3	4	5	6	7	8	9	10	11
P-value (ACF-based)	DMSO vs AICAR	0.0016	0.1381	0.2496	0.0073	0.4174	0.1584	0.0685	0.3859	0.0278	0.3052	--
	DMSO cv CC	0.1127	0.3077	0	0.0552	0.1088	0.3805	0.2276	0.0297	0.0040	0.2022	--
	AICAR vs CC	0.0452	0.0751	0.0253	0.2812	0.1706	0.2478	0.0076	0.0419	0.0072	0.2953	--
P-value (Deep Features based	DMSO vs AICAR	0.0036	0.0949	0.4475	0.0521	0.1856	0.1381	0.0295	0.4619	0.0312	0.0037	0.0612
	DMSO cv CC	0.0137	0.2778	0.1836	0.0503	0.0107	0.4764	0.1844	0.0644	0.0247	0.2416	0.0007
	AICAR vs CC	0.1918	0.0555	0.2185	0.4021	0.1666	0.1376	0.0001	0.1630	0.0004	0.0768	0.0116

Table 4.6: The p-values for statistical analysis of cell proportion in fine-grained phenotypes in Cluster 7 by ACF-based clustering.

ACF-based Clustering on Cluster 7		1	2	3
Arp2/3	CK689 vs CK666	0.0339	0.2735	0.0242
VASP	DMSO vs CyD50	0.0284	0.4669	0.3157
	DMSO vs CyD100	0.0284	0.1624	0.0863
	CyD50 vs CyD100	0.5100	0.2503	0.2286
Myosin	DMSO/Bleb	0.4033	0.4806	0.3245
AMPK	DMSO vs AICAR	0.2082	0.0218	0.0106
	DMSO vs CC	0.2583	0.4299	0.4547
	AICAR vs CC	0.0431	0.0711	0.0154

Table 4.7: The p-values for statistical analysis of cell proportion in fine-grained phenotypes in Cluster 7 by Deep Features-based clustering

Deep Features-based Clustering on Cluster 7		1	2	3	4
Arp2/3	CK689 vs CK666	0.0846	0.0278	0.0023	0.4142
VASP	DMSO vs CyD50	0.0353	0.3042	0.0686	0.0597
	DMSO vs CyD100	0.1595	0.0125	0.0062	0.6442
	CyD50 vs CyD100	0.1481	0.1029	0.2276	0.1032
Myosin	DMSO/Bleb	0.2910	0.4322	0.2522	0.4169
AMPK	DMSO vs AICAR	0.2602	0.0588	0.0312	0.4971
	DMSO vs CC	0.4664	0.2762	0.0860	0.0751
	AICAR vs CC	0.2220	0.0167	0.0019	0.1412

Table 4.8: The p-values for statistical analysis of cell proportion in fine-grained phenotypes in Cluster 2 by ACF-based clustering

ACF-based Clustering on Cluster 2		1	2	3	4	5
Arp2/3	CK689 vs CK666	0.4796	0.1668	0.0473	0.4497	0.2041
VASP	DMSO vs CyD50	0.0936	0.0964	0.2439	0.0033.	0.0179
	DMSO vs CyD100	0.0462	0.1551	0.4732	0.0059	0.0027
	CyD50 vs CyD100	0.4993	0.0206	0.2690	0.2594	0.4378
Myosin	DMSO/Bleb	0.0090	0.0954	0.4483	0.0341	0.4532
AMPK	DMSO vs AICAR	0.1045	0.2111	0.3406	0.3749	0.4564
	DMSO vs CC	0.4313	0.2470	0.0053	0.3818	0.3428
	AICAR vs CC	0.1302	0.4352	0.0361	0.2793	0.3375

Table 4.9: The p-values for statistical analysis of cell proportion in fine-grained phenotypes in Cluster 2 by Deep Features-based clustering

Deep Features-based Clustering on Cluster 2		1	2	3	4	5	6
Arp2/3	CK689 vs CK666	0.5418	0.1376	0.0552	0.1084	0.3542	0.1160
VASP	DMSO vs CyD50	0.1253	0.1833	0.1362	0.4651	0.3969	0.0036
	DMSO vs CyD100	0.1196	0.1342	0.0002	0.0673	0.0564	0.0486
	CyD50 vs CyD100	0.4142	0.0455	0.0693	0.0839	0.0486	0.8020
Myosin	DMSO/Bleb	0.0003	0.2525	0.2720	0.2792	0	0.1268
AMPK	DMSO vs AICAR	0.0233	0.2258	0.2879	0.4337	0.2499	0.2277
	DMSO vs CC	0.4335	0.0217	0.3894	0.1656	0.4636	0.4247
	AICAR vs CC	0.0206	0.0876	0.2611	0.1504	0.1859	0.2203

Figure Legends

Figure 4.1. Schematic Representation of the Analytical Steps of DeepHACKS (a) Cell-mask labeled time-lapse movies of the leading edge of a migrating PtK1 cell were taken at 5 seconds per frame for 1000 seconds, and then probing widows (500 x 500 nm) are generated to track the cell edge movement to generate the protrusion velocities. **(b)** all the protrusion events are detected and the protrusion onsets, which is the window begins to protrusion is defined to align the time series of protrusion velocities. The samples whose length is larger than 51 frames after protrusion onset are truncated. **(c-e) ACF-based clustering. (c)** For each sample**,** among the samples with similar length, shorter samples were padded with random noise to make the samples equal. SAX is used to represent the raw time series of velocity to eliminate the time dependence and noise. Then ACF-based distance is calculated for similarity measurement. (d) the similarity measurement is calculated for each sample across the whole dataset to generate partial similarity matrix. (e) graph-based clustering is applied on the partial similarity matrix to generate different clusters, called coarse-grained labels. **(f-g) Deep Feature Clustering. (f)** the raw velocity is rescaled nonlinearly to eliminate the effect of large velocity magnitude. (g) The guidance Bi-LSTM autoencoder structure. After training the model, the output of hidden layers is extracted for further analysis, called 'Deep Features'. Graph-based clustering is applied on the deep features to get distinguished phenotypes. **(h)** Using the features extracted from ACF or deep features, the interested clusters are further divided into several fine-grained phenotypes. **(i)** The phenotypes susceptible to pharmacological perturbations are identified based on the difference of velocity profiles from control and drug-treatment experiments. The drug-sensitive phenotypes are further evaluated by cluster proportion change quantitatively.

Figure 4. 2 guidance Bi-LSTM autoencoder for deep features extraction and visualization (a) the proposed neural network structure of guided Bi-LSTM autoencoder including encoder, decoder and MLP

classifier**. (b-d)** (b)The total loss curve of training and validation during the training process. (c) The reconstruction loss curve of training and validation. (d) the classification loss curve of training and validation. **(e)** The explained variance of PCA analysis. **(f)** The heatmap of total rescaled velocity dataset (g) The output of our framework as the reconstruction of total rescaled velocity dataset. **(h-j)** U-map visualization of deep features learned from guided Bi-LSTM autoencoder(h), Autoencoder (i) and only Classification (j). Here, the coarse-grained labeled were labelled here.

Figure 4.3 Subcellular Protrusion Phenotypes Revealed on paired experiments CK689/CK666. (a-d) The Subcellular Protrusion phenotypes revealed by ACF-based clustering. **(a)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(b-c)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (b) CK689 control and (c) CK666 drug-treatment experiment separately. **(d)** Comparison of the proportion of each cluster between CK689 (25 µM, inactive control compound) and CK666 (25 µM). The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. **(e-h) The subcellular protrusion phenotypes revealed by Deep Feature-based clustering**. **(e)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(f-g)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (f) CK689 control and (g) CK666 drug-treatment experiment separately. **(h)** Comparison of the proportion of each cluster between CK689 (25 µM, inactive control compound) and CK666 (25 µM). The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling.

Figure 4.4 Subcellular Protrusion Phenotypes Revealed on paired experiments DMSO/CyD50/CyD100. (a-d) The Subcellular Protrusion phenotypes revealed by ACF-based clustering. **(a)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(b-d)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (b) DMSO as a control experiment and (c) CyD50 (50 nM) drug-treatment experiment (d) CyD100 (100nM) drug treatment experiment separately. **(e)** Dose-response of the proportions of clusters to Cytochalasin D (50 or 00 nM) treated cells. The error bars indicate 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. **(f-j) The subcellular protrusion phenotypes revealed by Deep Feature-based clustering**. **(f)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(g-i)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (g) DMSO control and (h) CyD50 (50 nM) or (i) CyD100

(100 nM) drug-treatment experiment separately. **(j)** Comparison of the proportion of each cluster between DMSO and CyD (50 nM or 100 nM). The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling.

Figure 4.5 Subcellular Protrusion Phenotypes Revealed on paired experiments DMSO/Bleb. (a-d) The Subcellular Protrusion phenotypes revealed by ACF-based clustering. **(a)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(b-c)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (b) DMSO control and (c) Blebbistatin (20 μM) drug-treatment experiment separately. **(d)** Comparison of the proportion of each cluster between DMSO and Bleb (20 μM). The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. **(e-h) The subcellular protrusion phenotypes revealed by Deep Feature-based clustering**. **(e)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(f-g)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (f) DMSO control and (g) Bleb drug-treatment experiment separately. **(h)** Comparison of the proportion of each cluster between DMSO and Bleb. The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling.

Figure 4.6 Subcellular Protrusion Phenotypes Revealed on paired experiments DMSO/AICAR/CC. (a-d) The Subcellular Protrusion phenotypes revealed by ACF-based clustering. **(a)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(b-d)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (b) DMSO as a control experiment and (c) AICAR (1mM) drug-treatment experiment (d) CC drug treatment experiment separately. **(e)** Dose-response of the proportions of clusters to AICAR and CC treated cells. The error bars indicate 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. **(f-j) The subcellular protrusion phenotypes revealed by Deep Feature-based clustering**. **(f)** Ensemble averaged velocity time series of samples in each cluster (from left to right on 1st row: cluster 1 to cluster 5; from left to right on 2nd row: cluster 6 to cluster 10). **(g-i)** A t-SNE plot of partial similarity distance of protrusion velocity time series overlaid with cluster assignments for (g) DMSO control and (h) AICAR and (i) CC drug-treatment experiment separately. **(j)** Comparison of the proportion of each cluster among DMSO, AICAR and CC. The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling.

Figure 4.7 t-SNE visualization on Cluster 2 and Cluster 7 from ACF-based clustering and Deep Features-based clustering. (a) t-SNE visualization on cluster 2 using ACF-based clustering. **(b)** t-SNE

visualization on cluster 7 using ACF-based clustering. **(c)** t-SNE visualization on cluster 2 using Deep Features-based clustering. **(d)** t-SNE visualization on cluster 7 using Deep Features-based clustering.

Figure 4.8 Fine-grained Protrusion Phenotypes in Acceleration Protrusion (Cluster 7) revealed by ACF-based Clustering. (a-d) Three clusters ensemble averaged velocity time series of sample identified in each paired experiment. (a) CK689/CK666 (b) DMSO/CyD50/CyD100 (c) DMSO/Bleb (d) DMSO/AICAR. **(e-h)** Comparison of the proportion in each paired experiment. The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. (e) CK689/CK666 (f) DMSO/CyD50/CyD100 (g) DMSO/Bleb (h) DMSO/AICAR.

Figure 4.9 Fine-grained Protrusion Phenotypes in Acceleration Protrusion (Cluster 7) revealed by Dee Features-based Clustering. (a-e) Three clusters ensemble averaged velocity time series of sample identified in each paired experiment. (a) Pooled average across different experiment (b) CK689/CK666 (c) DMSO/CyD50/CyD100 (d) DMSO/Bleb (e) DMSO/AICAR. **(f-i)** Comparison of the proportion in each paired experiment. The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. (f) CK689/CK666 (g) DMSO/CyD50/CyD100 (h) DMSO/Bleb (i) DMSO/AICAR.

Figure 4.10 Fine-grained Protrusion Phenotypes in Bursting Protrusion (Cluster 2) revealed by ACF-based Clustering. (a-d) Three clusters ensemble averaged velocity time series of sample identified in each paired experiment. (a) CK689/CK666 (b) DMSO/CyD50/CyD100 (c) DMSO/Bleb (d) DMSO/AICAR. **(e-h)** Comparison of the proportion in each paired experiment. The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. (e) CK689/CK666 (f) DMSO/CyD50/CyD100 (g) DMSO/Bleb (h) DMSO/AICAR.

Figure 4.11 Fine-grained Protrusion Phenotypes in Bursting Protrusion (Cluster 2) revealed by Dee Features-based Clustering. (a-e) Three clusters ensemble averaged velocity time series of sample identified in each paired experiment. (a) Pooled average across different experiment (b) CK689/CK666 (c) DMSO/CyD50/CyD100 (d) DMSO/Bleb (e) DMSO/AICAR. **(f-i)** Comparison of the proportion in each paired experiment. The error bars indicated 95% confidence interval of the mean of the cluster proportions. * ($p < 0.05$) indicates the statistical significance by bootstrap sampling. (f) CK689/CK666 (g) DMSO/CyD50/CyD100 (h) DMSO/Bleb (i) DMSO/AICAR.

Literature Cited

1. Orth, Antony, et al. "Gigapixel multispectral microscopy." *Optica* 2.7 (2015): 654-662.
2. Orth, Antony, Diane Schaak, and Ethan Schonbrun. "Microscopy, meet big data." *Cell systems* 4.3 (2017): 260-261.

3. Small, J. V., Stradal, T., Vignal, E. & Rottner, K. The lamellipodium: where motility begins. Trends Cell Biol 12, 112-120 (2002).
4. Pankov, R. *et al.* A Rac switch regulates random versus directionally persistent cell migration. *J Cell Biol* 170, 793-802, doi:10.1083/jcb.200503152 (2005).
5. Lauffenburger, D. A. & Horwitz, A. F. Cell migration: a physically integrated molecular process. *Cell* 84, 359-369 (1996).
6. Lee, K. et al. Functional hierarchy of redundant actin assembly factors revealed by fine-grained registration of intrinsic image fluctuations. Cell Syst 1, 37-50, (2015).
7. Wang, Chuangqi, et al. "Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging." Nature communications 9.1 (2018): 1688.
8. Zhang, Fan, and Patrick Flaherty. "Variational inference for rare variant detection in deep, heterogeneous next-generation sequencing data." BMC bioinformatics 18.1 (2017): 45.
9. Robinson, Peter N. "Deep phenotyping for precision medicine." Human mutation 33.5 (2012): 777-780.
10. San-Miguel, Adriana, et al. "Deep phenotyping unveils hidden traits and genetic relations in subtle mutants." Nature communications 7 (2016): 12990.
11. Schneider, C. A.; Rasband, W. S. & Eliceiri, K. W. (2012), "NIH Image to ImageJ: 25 years of image analysis"
12. Carpenter, Anne E., et al. "CellProfiler: image analysis software for identifying and quantifying cell phenotypes." Genome biology 7.10 (2006):
13. Kamentsky, Lee, et al. "Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software." Bioinformatics 27.8 (2011): 1179-1180.
14. Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006.
15. Hastie, Trevor, et al. "The elements of statistical learning: data mining, inference and prediction." The Mathematical Intelligencer 27.2 (2005): 83-85.
16. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.
17. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
18. Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
19. Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.
20. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
21. Shen, Dinggang, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." Annual review of biomedical engineering 19 (2017): 221-248.
22. Erick Moen, et al. “Deep learning for cellular image analysis” nature method (2019).
23. Hinton, G.E., and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. science 313, 504-507.
24. Saul, L.K., and Roweis, S.T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of machine learning research 4, 119-155.
25. Machacek, M., and Danuser, G. (2006). Morphodynamic profiling of protrusion phenotypes. Biophysical journal 90, 1439-1452.
26. Machacek, M., Hodgson, L., Welch, C., Elliott, H., Pertz, O., Nalbant, P., Abell, A., Johnson, G.L., Hahn, K.M., and Danuser, G. (2009). Coordination of Rho GTPase activities during cell protrusion. Nature 461, 99.
27. Antonello, Z.A., Reiff, T., Ballesta‐Illan, E., and Dominguez, M. (2015). Robust intestinal homeostasis relies on cellular plasticity in enteroblasts mediated by miR‐8–Escargot switch. The EMBO journal 34, 2025-2041.

28. Morikawa, Y., Zhang, M., Heallen, T., Leach, J., Tao, G., Xiao, Y., Bai, Y., Li, W., Willerson, J.T., and Martin, J.F. (2015). Actin cytoskeletal remodeling with protrusion formation is essential for heart regeneration in Hippo-deficient mice. Sci Signal 8, ra41-ra41.
29. Ioannou, M.S., Bell, E.S., Girard, M., Chaineau, M., Hamlin, J.N., Daubaras, M., Monast, A., Park, M., Hodgson, L., and McPherson, P.S. (2015). DENND2B activates Rab13 at the leading edge of migrating cells and promotes metastatic behavior. J Cell Biol 208, 629-648.
30. Liu, Y.-h., Jin, J.-l., Wang, Y.-z., Tan, Y., Zhou, Y.-y., Peng, T., Li, F., Liang, W.-d., Chartrand, P., and Jiang, Y.-y. (2016). Protrusion-localized STAT3 mRNA promotes metastasis of highly metastatic hepatocellular carcinoma cells in vitro. Acta Pharmacologica Sinica 37, 805.
31. Taniuchi, K., Furihata, M., Hanazaki, K., Saito, M., and Saibara, T. (2014). IGF2BP3-mediated translation in cell protrusions promotes cell invasiveness and metastasis of pancreatic cancer. Oncotarget 5, 6832.
32. Leithner, A., Eichner, A., Müller, J., Reversat, A., Brown, M., Schwarz, J., Merrin, J., de Gorter, D.J., Schur, F., and Bayerl, J. (2016). Diversified actin protrusions promote environmental exploration but are dispensable for locomotion of leukocytes. Nature cell biology 18, 1253.
33. Lin, J., Keogh, E., Lonardi, S. & Chiu, B. A symbolic representation of time series, with implications for streaming algorithms. Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM, 2-11 (2003).
34. Rosvall, Martin, and Carl T. Bergstrom. "Maps of random walks on complex networks reveal community structure." Proceedings of the National Academy of Sciences 105.4 (2008): 1118-1123.
35. Pierpaolo, D. & Maharaj, E. A. Autocorrelation-based fuzzy clustering of time series. Fuzzy Sets and Systems 160, 3565-3589 (2009).
36. Laurens van der, M. & G., H. Visualizing data using t-SNE Journal of Machine Learning Research 9, 2579-2605 (2008).
37. Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of machine learning research 11.Dec (2010): 3371-3408.
38. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997
39. Davies, David L.; Bouldin, Donald W. (1979). "A Cluster Separation Measure". IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-1 (2): 224–227.
40. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53-65 (1987).
41. McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." arXiv preprint arXiv:1802.03426 (2018).
42. Huang, N. E. et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 454 (1998).
43. Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).
44. Caron, Mathilde, et al. "Deep clustering for unsupervised learning of visual features." Proceedings of the European Conference on Computer Vision (ECCV). 2018.
45. Wager, Stefan, Alexander Blocker, and Niall Cardin. "Weakly supervised clustering: Learning fine-grained signals from coarse labels." The Annals of Applied Statistics 9.2 (2015): 801-820.
46. Le, Lei, Andrew Patterson, and Martha White. "Supervised autoencoders: Improving generalization performance with unsupervised regularizers." Advances in Neural Information Processing Systems. 2018.
47. Shafqat-Abbasi, Hamdah, et al. "An analysis toolbox to explore mesenchymal migration heterogeneity reveals adaptive switching between distinct modes." Elife 5 (2016): e11384.

48. Gordonov, Simon, et al. "Time series modeling of live-cell shape dynamics for image-based phenotypic profiling." Integrative Biology 8.1 (2015): 73-90.
49. Verkhovsky, Alexander B. "The mechanisms of spatial and temporal patterning of cell-edge dynamics." Current opinion in cell biology 36 (2015): 113-121.
50. Ananthakrishnan, Revathi, and Allen Ehrlicher. "The forces behind cell movement." International journal of biological sciences 3.5 (2007): 303.
51. Suraneni, Praveen, et al. "A mechanism of leading-edge protrusion in the absence of Arp2/3 complex." Molecular biology of the cell 26.5 (2015): 901-912.
52. Gough, Albert, et al. "Biologically relevant heterogeneity: Metrics and practical insights." SLAS DISCOVERY: Advancing Life Sciences R&D 22.3 (2017): 213-237.

Chapter 5

Conclusions and Outlook

This dissertation focuses on the development of machine learning pipelines that address challenges in deconvoluting the protrusion heterogeneity at the subcellular level using live cell imaging dataset. Also, I explored the possibility of transfer deep learning to extract efficient features from cell imaging for molecular diagnostics

5.1 Summary of contributions

In this dissertation, we addressed two main problems: cell imaging analysis and subcellular protrusion heterogeneity. Technically, we developed several pipelines from traditional machine learning methods to deep learning (feature representation). Using these proposed pipelines, I demonstrated the success to address our challenging problems by improve the performance or discover the novel phenotypes with specific biological mechanisms. The contributions of this dissertation could be divided into two parts: deep learning applications in cell imaging process and subcellular protrusion heterogeneity using time series analysis.

5.1.1 Cell imaging process

Cell imaging process is widely used and plays a vital role in quantitative biological data analysis. As the first step, it's important to extract useful information with high quality for further analysis. Therefore, how to extract features will become fundamental and important. However, it's challenging to design a general feature from different and complex microscopy datasets since different microscopy techniques have significant and different characteristics. For example, there is high noise in florescence microscopy but in phase-contrast microscopy, the 'halo' effect makes it difficult to extract information. Here, I utilized the deep learning techniques to automatically extract feature for molecular diagnostic. Specifically, in chapter two, for the hologram images from lens-free digital in-line hologramphy (LDIH), currently, there is no efficient manual-design features. Using the feature learned from VGG16 in ImageNet dataset, I demonstrated to extract efficient features for molecular diagnostic.

5.1.2 Subcellular protrusion heterogeneity

Heterogeneity widely exists in biological systems. Deconvolution of cell protrusion is crucial to understanding the biology of cellular movement. However, typical ensemble average techniques conceal the heterogeneity and fail to discover novel phenomena. Here, we proposed the cutting-edge machine learning pipelines to deal with this heterogeneity as follows:

1) For the long protrusion event with equal length, we proposed a machine learning pipeline called HACKS to dissect protrusion heterogeneity from live cell imaging. It can identify hidden patterns

from a complex and heterogeneous velocity time series data (phenotype) and reveal the specific effects of drug treatment.

2) For the short protrusion event with variable length, we developed a deep learning pipeline to learn features integrating the prior information to discover fined-grained phenotypes. I can identify rare subcellular phenotypes specifically sensitive to molecular and environmental perturbations.

Here, I demonstrated how important it is to dissect the heterogeneity for novel phenotype discovery and drug characterization.

5.2 Future Directions

Cell migration is a complex phenomenon and can be divided into four steps: leading edge extension, nascent adhesion generation, cell body crawling and de-attaching at the rear part. In the first step, the motility in leading edge extension comprise cell protrusion and cell retraction iteratively. In this dissertation, we only explored the heterogeneity of subcellular protrusion and demonstrated the novel phenotypes with specific mechanism of actin regulator coordination. In future, there are several potential research directions worth to explore.

5.2.1 Challenging and Opportunity of protrusion heterogeneity

One critical challenge for the current pipeline to dissect the subcellular protrusion heterogeneity is to accurately extract time series protrusion activities tracking each small region called ‘probing window’ along the cell boundary. It requires accurate and robust cell boundary segmentation and precise window tracking. Therefore, more accurate and robust boundary segmentation methods are expected. Furthermore,for the tracking probing windows, we directly applied the method proposed by Machacek[1]without any modification, which works well for the short-term window tracking and have some potential issues for long-term tracking. The advanced local sampling method is expected to track the edge more accurately. One possible solution will transform the window tracking problem into an optimization problem to find the optimal boundary locations by maximizing consistent assignment of pixels into the windows as an objective function, utilizing the direction of velocity as a constraint for each pixel tracking.

Another challenge for quantifying the rare subcellular phenotypes specifically sensitive to molecular and environmental perturbations is to purify the phenotypes by eliminating the noised samples. Especially, when we focused on the fine-grained phenotypes, which is only 1-5 percentage among the whole samples, even small proportion of noised samples will affect statistical significance. One possible solution is to introduce mapping methods using canonical correlation analysis (CCA)[2], which will eliminate the general effect and experimental drift from different experiments, to align the samples from both control and drug-treatment experiments. If in some mapping dimensions without no clear drift, another solution is to integrate the classification or criteria measurement from soft clustering to assign the possibility to

each sample for each phenotype. Then we can trim samples with low possibility, which could decontaminate the phenotypes for better quantification.

5.2.2 From protrusion to retraction at the leading edge

Furthermore, cell retraction makes significant contribution to cell motility at the leading edge. Dissecting the heterogeneity of cell retraction and elucidating the relationship between protrusion and retraction will help the researcher to understand cell motility better. During our analysis, we observed that cell retraction is also heterogeneous at the subcellular level. We believe that our pipeline could directly apply to the phenomenon of cell retraction without modification. To elucidate the regulator mechanism involved in cell retraction, it is necessary to design the experiments where we monitor retraction related regulators in addition to Arp2/3 or VASP for further research.

Moreover, the relationship between protrusion and retraction is also an interested topic to explore. One possible solution is a bottom-up approach, which first dissects the heterogeneity of protrusion and retraction separately and then discover protrusion-retraction pattern by evaluation the subcellular phenotypes between protrusion and retraction. Another possible solution is to extract the protrusion-retraction cycle time series data and then adjust our pipeline to extract efficient information for further cycle phenotype identification. Furthermore, we can apply time-series modeling methods[3] and then use the modeling parameters for further analysis.

Furthermore, cell heterogeneity is a fundamental property of complex cell systems. Studying the mechanism of heterogeneity is essential to understand and predict cellular response. Since so many internal and external factors such as local environments are involved to affect cell states at the cellular level, it is challenging to associate the role of specific factors with the different cellular states directly. However, at the subcellular level, we successfully associated the subcellular phenotypes with regulator dynamics. Therefore, utilizing the bottom-up strategy, it seems promising to interpret the cellular states based on the combination of subcellular phenotypes. For example, after we get the protrusion and retraction phenotypes for different cells, it is possible to understand the difference based on the different proportion of protrusion and retraction phenotypes. Finally, Studying the subcellular protrusion or retraction heterogeneity will allow us to understand the cellular heterogeneity of cell migration.

5.2.3 From leading edge to inside of cell boundary

Beyond the leading edge, ranging from 2 to 10 μm localized at the front region, it is well accepted that other effectors inside the cell body are also involved in cell protrusion. For example, the traction force generated from focal adhesion and membrane tension also could play roles in the heterogeneity of cell protrusion.[4,5] New experimental and computational pipeline is expected to find the hidden mechanism of traction force or other adhesion regulators inside cell body.

Literature cited

1. Machacek, M., and G. Danuser. 2006. Morphodynamic Profiling of Protrusion Phenotypes. *Biophys. J.* 90:1439-1452.
2. Hardoon, David R., Sandor Szedmak, and John Shawe-Taylor. "Canonical correlation analysis: An overview with application to learning methods." *Neural computation* 16.12 (2004): 2639-2664.
3. Kim, Yeesock, Hee June Choi, and Kwonmoo Lee. "Subcellular Time Series Modeling of Heterogeneous Cell Protrusion." bioRxiv (2018): 429118.
4. Pontes, Bruno, et al. "Membrane tension controls adhesion positioning at the leading edge of cells." J Cell Biol 216.9 (2017): 2959-2977.
5. Parsons, J. Thomas, Alan Rick Horwitz, and Martin A. Schwartz. "Cell adhesion: integrating cytoskeletal dynamics and cellular tension." *Nature reviews Molecular cell biology*11.9 (2010): 633.

SUPPLEMENTARY NOTES in Chapter 3

Suppementary Note 3.1: Evaluation of the role of each component of time series clustering

Our time series clustering primarily consisted of SAX (dimensional reduction), ACF distance (dissimilarity measure with Autocorrelation Function) and Density Peak clustering. To identify the role of each component in our pipeline, we systematically tested the performance of the clustering results by replacing each component with other methods. First, without SAX for dimensional reduction, two different distance dissimilarity measures, ACF and Euclidean distances were tested. When Euclidean distance was applied without SAX, we could not find distinct cluster structure using the Density Peak clustering (Supplementary Fig. 1a), and we were not able to extract periodic and accelerating protrusions from the de-noised velocity samples (Supplementary Fig. 1a-c). When we applied community detection using Euclidean distance without SAX, we still could not identify periodic and

accelerated protrusions (Supplementary Fig. 1d-f). When the ACF distance was used without SAX, we were able to identify accelerating protrusions (Supplementary Fig. 1g-j) using Density Peak clustering, but we could not find the full spectrum of the periodic protrusion patterns (Supplementary Fig. 1j). However, we were able to identify periodic and accelerating protrusion using Community Detection method, when the ACF distance was used without SAX (Supplementary Fig. 1k-m). Furthermore, when we tested the clustering using SAX and its Euclidean distance (the lower bound of Euclidean distance in SAX), we were not able to identify distinct temporal patterns of protrusion velocity (Supplementary Fig. 1n-q). Taken together, we concluded that the ACF distance is the most important factor which allowed us to extract the periodic and accelerating protrusion.

When we used ACF distance with SAX, the Community Detection method was able to identify both accelerating protrusion and three periodic protrusions when the number of clusters was set to be six (Supplementary Fig. 2 h-i). However, it could not identify one periodic pattern when the number of clusters is set to be five (Supplementary Fig. 2 f-g). Since the silhouette values in Supplementary Fig. 2f,h is larger than those in Supplementary Fig. 1k, SAX seems to play a role in making cluster tighter and cleaner by reducing the local data fluctuation. Using ACF distance with SAX, density Peaks can extract both acceleration protrusion and three periodic protrusions (Fig. 2).

Finally, we also applied the conventional K-means clustering algorithm to our data using ACF distance with SAX (Supplementary Fig. 2a-e). First, we determine the optimal number of clusters to be seven using both Davies–Bouldin Index and Silhouette criteria (Supplementary Fig. 2a). Due to the random initialization of cluster centers, we showed two clustering results from K-means (Supplementary Fig. 2b-c and Supplementary Fig. 2d-e), where both acceleration protrusion and periodic protrusions were identified. Therefore, the combination of ACF and SAX, three clustering methods (K-means, Density Peaks, and Community Detection) were able to isolate the acceleration protrusion and three periodic protrusions patterns. However, in K-means and Community Detection, the optimal number of clusters is not easily determined. To delineate both periodic and accelerating protrusions, we needed to use the larger number of clusters, which resulted in small numbers of samples in each cluster. Density Peaks not only provided the information to determine the number of clusters but also effectively discovers both periodic and accelerating protrusions with a relatively smaller number of clusters. Moreover, the density peak is a deterministic method while K-means and community detection are involved in the random initiation and random walk. Therefore, the Density Peak can produce more reproducible results. These results were summarized in Supplementary Table 2.

Suppementary Note 3.2: The correlation between early Arp2/3 and late protrusion velocities

The significant and strong instantaneous correlation between VASP and the protrusion velocity begins to appear 100 seconds after protrusion onset (Fig. 4h, Cluster III), along with no correlation between actin and the protrusion velocity in Cluster III (Fig. 4f, Cluster III). Since the Arp3 intensity in Cluster III reached the maximum at approximately this time point (Fig. 3c, Cluster III), we hypothesized that

Arp2/3 in the early phase of Cluster III might play an important role. To test this idea, we integrated early Arp3 intensities between 0 to 50 seconds and correlated them with average protrusion velocities between 150 and 200 seconds in each cluster (Fig. 4i, Supplementary Fig. 5c). The Pearson's correlation coefficient Cluster III was significant (0.36, p = 0.0002, two tailed K-S test) and larger than those in other clusters. This is consistent with our hypothesis that Arp2/3 may be important in initiating cell protrusion in Cluster III.

Suppementary Note 3.3: Prediction of accelerating protrusion by VASP dynamics

Since VASP intensities were well correlated with protrusion velocities, we investigated whether VASP intensities contain sufficient information to predict protrusion phenotypes. To this end, we applied supervised learning approaches to further validate that VASP intensity time series can predict the acceleration phenotype. Using support vector machine (SVM), deep neural network (DNN), and random forest (RF), we built classifiers from the normalized intensities of actin, Arp3, and VASP to distinguish the non-accelerating (Clusters I and II-1/2/3) and accelerating (Cluster III) protrusion phenotypes. As a result, classifiers that were trained using VASP intensities could distinguish accelerating protrusions from non-accelerating protrusions with a significantly higher accuracy (Fig. 4j) and MCC (Matthews correlation coefficient) (Fig. 4k) than the classifiers trained using the actin and Arp3 intensities (p-values in Supplementary Table 4, two-tailed K-S test). Despite the modest classification accuracy, this difference suggests not only that VASP correlated with protrusion velocity but also that the correlation is strong enough to predict the accelerating protrusion phenotype.

Supplementary Table 3.1. Number of cells and probing windows used in the time series clustering analysis

	Global		Cluster I		Cluster II-1		Cluster I-2		Cluster II-3		Cluster III	
	#Video	#Window	#Video	#Window	#Video	#Window	#Video	#Window	#Video	#Window	#Video	#Window
All (=Velocity)	36	2756	34	764	35	367	35	625	36	674	33	326
Actin	10	934	10	215	10	134	10	255	10	245	10	85
Arp3	11	757	11	204	11	112	11	161	11	178	10	102
VASP	9	682	8	242	9	85	8	117	9	137	7	101
HaloTag	6	383	5	103	5	36	6	92	6	114	6	38

Supplementary Table 3.2. Summary of the results from the different combinations of algorithms in time series clustering

Method Combination				Persistent Pattern	Three Oscillating Patterns
DV	SAX	Distance	Clustering method		
Yes	No	ED	DP	No	No
Yes	No	ED	CD	No	No
Yes	No	ACF	DP	Yes	No
Yes	No	ACF	CD	Yes	Yes
Yes	Yes	app_ED	DP	No	No
Yes	Yes	ACF	Kmeans	Yes	Yes
Yes	Yes	ACF	CD	Yes	Yes
Yes	Yes	ACF	DP	Yes	Yes

DV: de-noised velocity
ED: Squared Euclidean Distance
app_ED: approximate Euclidean Distance in SAX
ACF: Squared Euclidean Distance based on Autocorrelation
SAX: Symbolic Aggregate Approximation
Kmeans: K-nearest mean method
DP: Density Peaks
CD: Community Detection

Supplementary Table 3.3. The p-values for statistical analyses of the maximum correlation coefficients from time-lag correlation analysis presented in Figure 3.4e All p-values were calculated by two-sample two-tailed Kolmogorov-Smirnov (KS) test.

	SVM			DNN			RF		
Protein pair	actin-Arp3	actin-VASP	Arp3-VASP	actin-Arp3	actin-VASP	Arp3-VASP	actin-Arp3	actin-VASP	Arp3-VASP
Accuracy	2.21×10^{-08}	5.06×10^{-30}	1.32×10^{-38}	0.047	1.27×10^{-28}	8.66×10^{-34}	0.003	2.96×10^{-32}	5.77×10^{-37}
MCC	2.21×10^{-08}	1.16×10^{-26}	3.70×10^{-36}	0.14	1.27×10^{-28}	9.30×10^{-31}	0.003	5.11×10^{-33}	2.33×10^{-35}

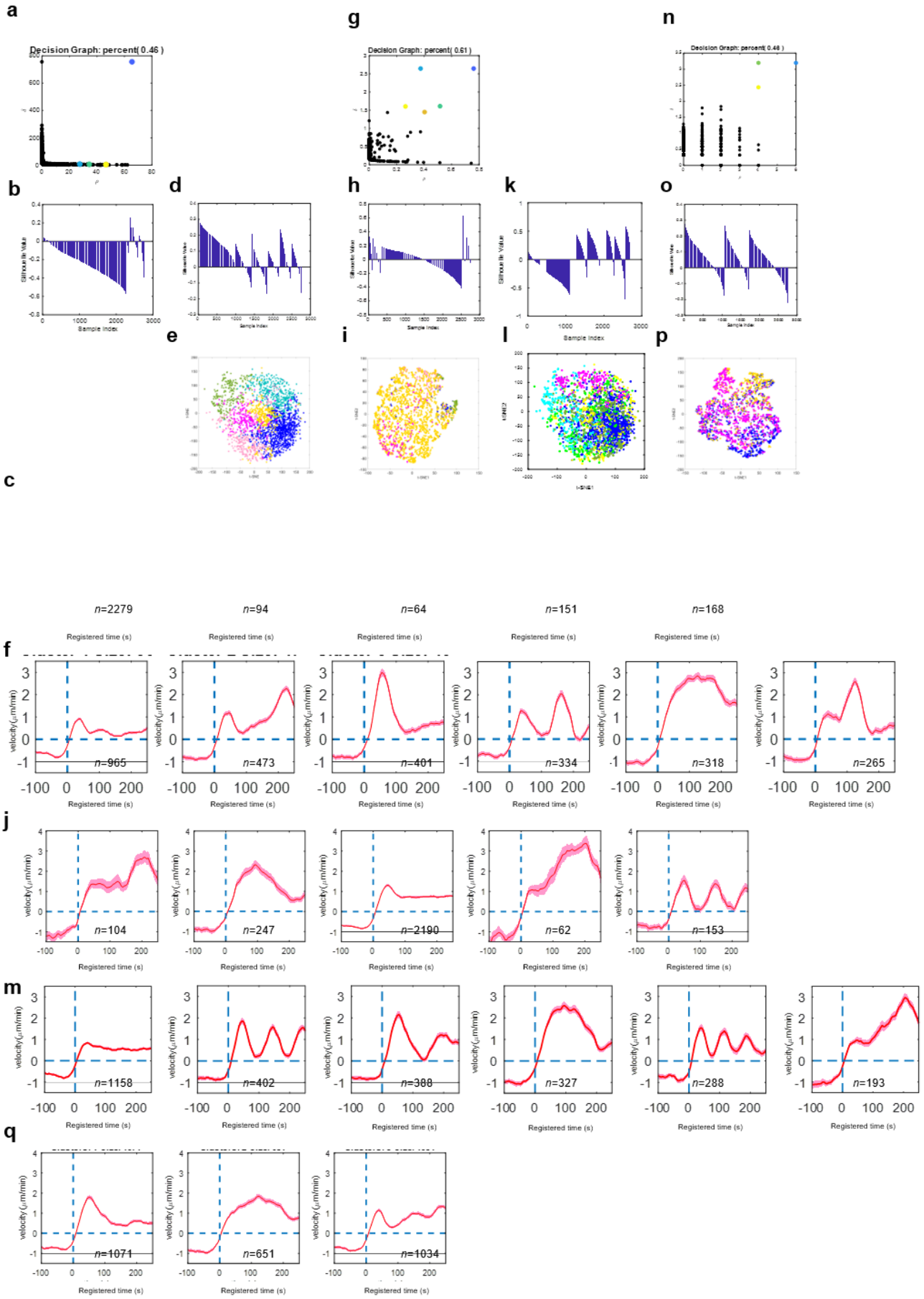
Supplementary Table 3.4. Statistical analyses of accuracy and Matthews correlation coefficients from the classification analysis presented in Figure 3.4j-k. The p-values of the differences in accuracy and Matthews correlation coefficients between the protein pairs from the classification analysis of Cluster III against Clusters I and II-1/2/3. All p-values were calculated by two-sample two-tailed Kolmogorov-Smirnov (KS) test.

	Cluster I	Cluster II-1	Cluster II-2	Cluster II-3	Cluster III
Actin-Arp3	3.47×10^{-3}	0.041	2.36×10^{-4}	3.64×10^{-8}	0.075
Arp3-VASP	8.77×10^{-17}	1.55×10^{-10}	3.90×10^{-9}	5.25×10^{-9}	3.87×10^{-14}
Actin-VASP	7.62×10^{-13}	1.16×10^{-5}	2.15×10^{-10}	2.86×10^{-3}	5.16×10^{-12}

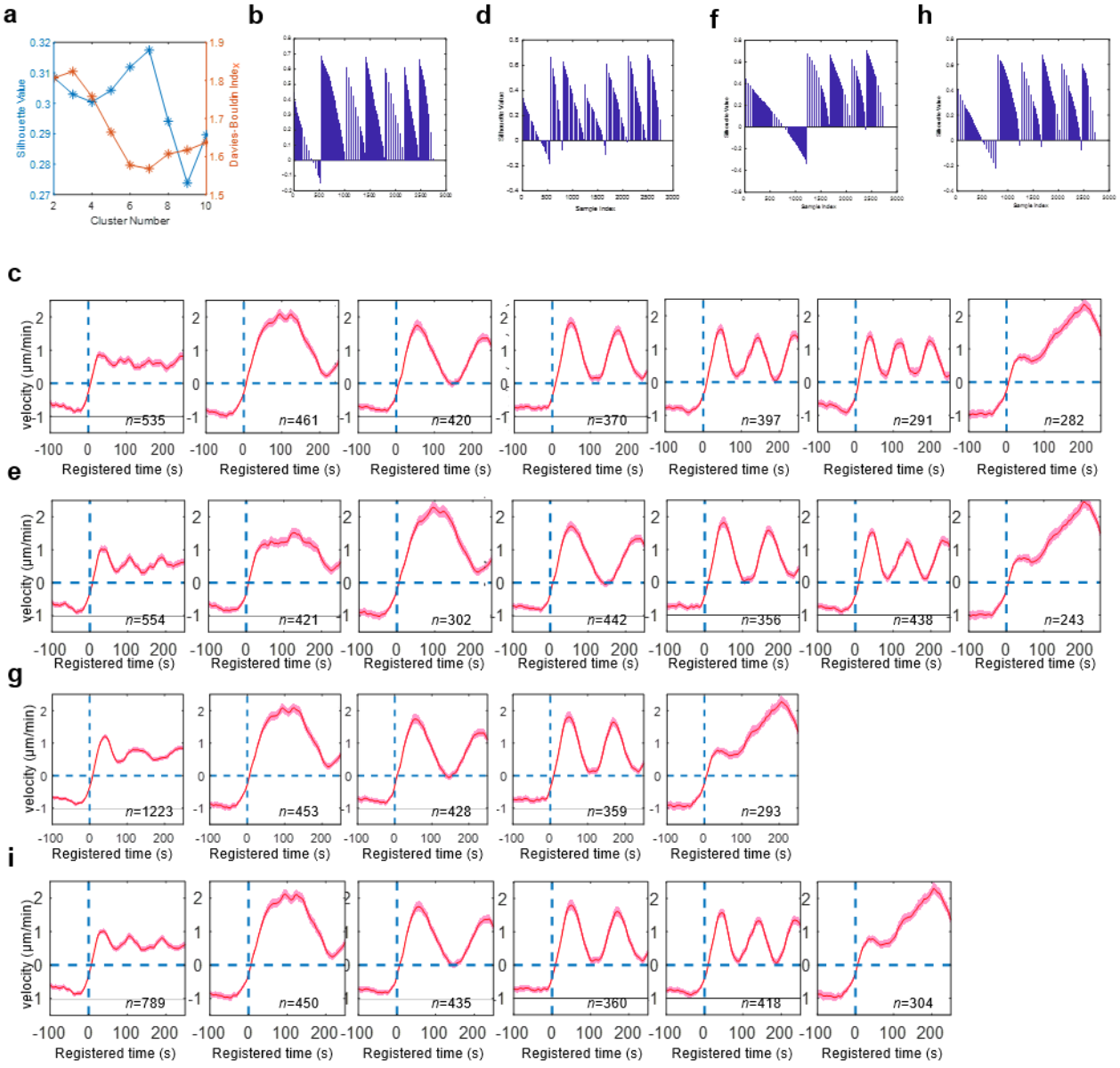
Supplementary Table 3.5. Hyperparameters used in the classification methods for each protein case. In Random Forest (RF), *N_estimaters* indicates the number of trees in the forest. *oob_score (true)* means to use out-of-bag samples to estimate the generalization accuracy. In Support Vector Machine (SVM), the parameter *C* shows the penalty of the error term while γ represents the kernel coefficient. In Deep Neural Network (DNN), there are three layers. Other non-specified parameters are set to the default parameters in Scikit-Learn and Keras package.

Methods	Hyperparameters	VASP	Arp2/3	Actin
RF	N_estimators	100		
	OOB_score	True		
SVM	C	1.93		
	γ	0.037	0.014	0.037
DNN	1st layer	Convolution, 3 filters with 3 taps, ReLU activation		
	2nd layer	Dense, 30 taps, 20% dropout, ReLU activation		
	3rd layer	Dense, 10 taps, 20% dropout, ReLU activation		

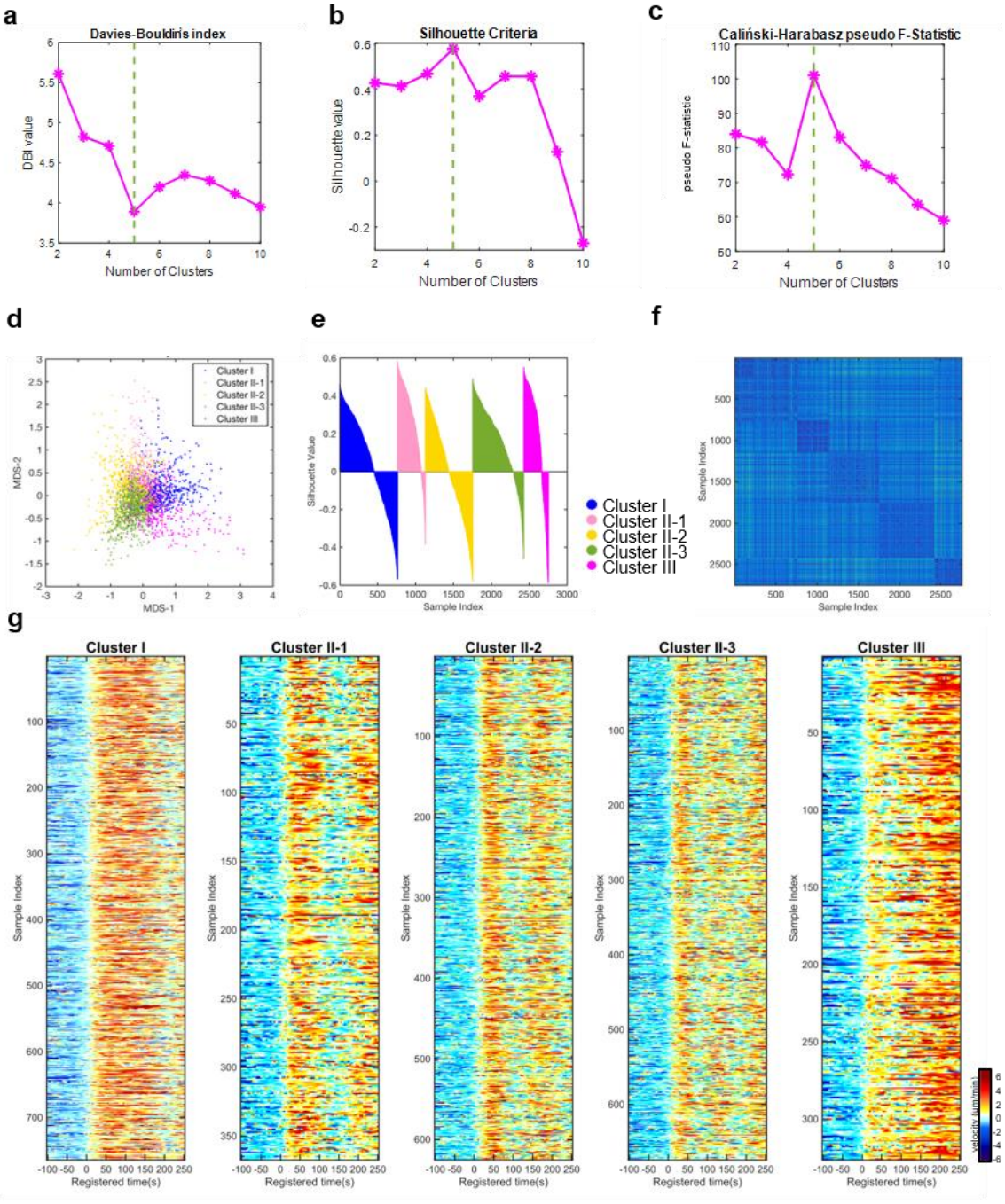
SVM: Support Vector Machine
DNN: Deep Neural Network
RF: Random Forest
MCC: Matthews Correlation Coefficient


a
g
n
Decision Graph: percent(0.46)
Decision Graph: percent(0.51)
Decision Graph: percent(0.48)
b
d
h
k
o
Silhouette Value
Sample Index
e
i
l
p
t-SNE1
t-SNE2
c
n=2279
n=94
n=64
n=151
n=168
f
n=965
n=473
n=401
n=334
n=318
n=265
j
n=104
n=247
n=2190
n=62
n=153
m
n=1158
n=402
n=388
n=327
n=288
n=193
q
n=1071
n=651
n=1034
velocity(μm/min)
Registered time (s)

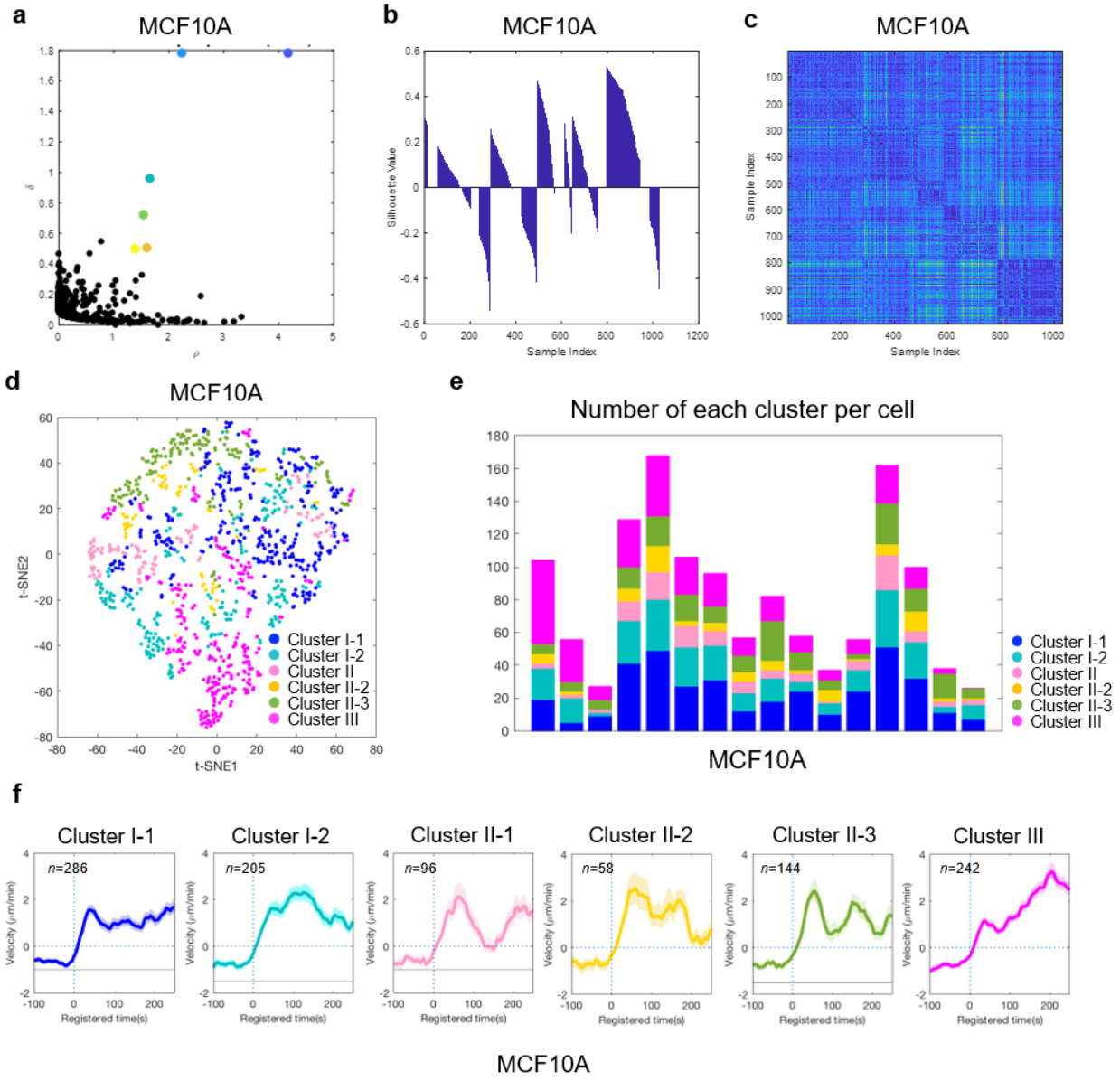
Supplementary Figure 3.1. Evaluating the role of ACF distance. (a-c) The density peak clustering results using Euclidean distance without SAX. (d-f) The community detection results using Euclidean distance without SAX. (g-j) The density peak clustering results using ACF distance without SAX. (k-m) The community detection results using ACF distance without SAX. (n-q) The density peak clustering results using app_ED distance with SAX. (a, g, n) density map, (b, d, h, k, o) silhouette plot, (e, i, l, p) t-SNE plot. (c, f, j, m, q) protrusion velocity profiles in each cluster. (*n:* the number of protrusion segments used for the analysis, images from 36 cells were used for the analysis.)



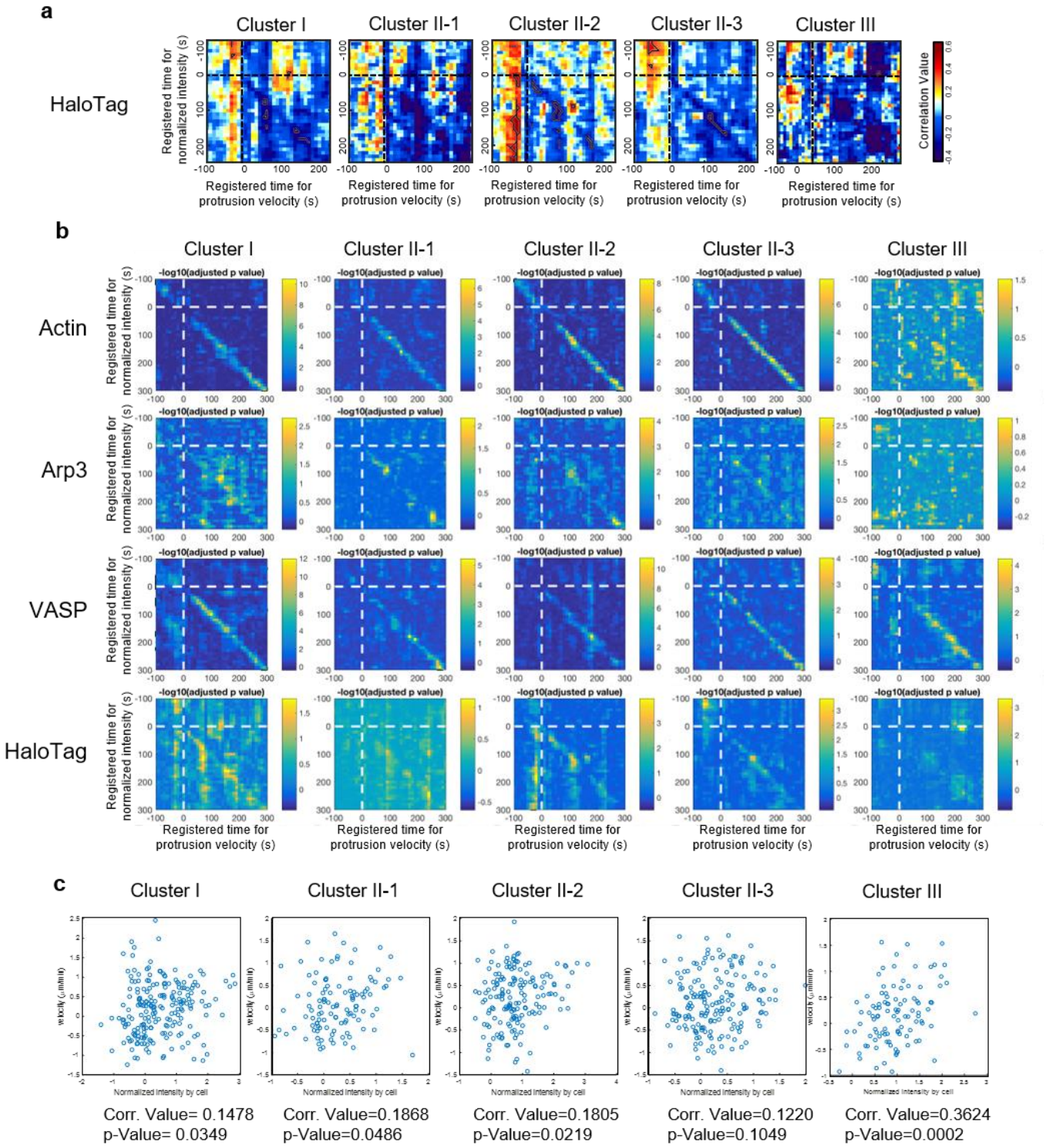
Supplementary Figure 3.2. Evaluating the roles of SAX and Density Peak clustering (a-e) The K-means clustering results using ACF distance with SAX. (a) Estimating the optimal number of clusters using DBI and silhouette value. (d-e) the result from another trials of K-mean. (f-i) The community detection results using ACF with SAX, (f-g) with five cluster condition and (h-i) with six cluster condition. (b, d, f, h) silhouette plots and (c, e g, i) protrusion velocity profiles of clusters. (*n*: the number of protrusion segments used for the analysis, images from 36 cells were used for the analysis.)


a
Davies-Bouldin's index
DBI value
Number of Clusters
b
Silhouette Criteria
Silhouette value
Number of Clusters
c
Caliński-Harabasz pseudo F-Statistic
pseudo F-statistic
Number of Clusters
d
Cluster I
Cluster II-1
Cluster II-2
Cluster II-3
Cluster III
MDS-2
MDS-1
e
Silhouette Value
Sample Index
Cluster I
Cluster II-1
Cluster II-2
Cluster II-3
Cluster III
f
Sample Index
Sample Index
g
Cluster I
Cluster II-1
Cluster II-2
Cluster II-3
Cluster III
Sample Index
Registered time(s)
velocity (um/min)
h
Number of samples of each clusters
Cluster I
Cluster II-1
Cluster II-2
Cluster II-3
Cluster III
Actin
Arp3
VASP
Halo

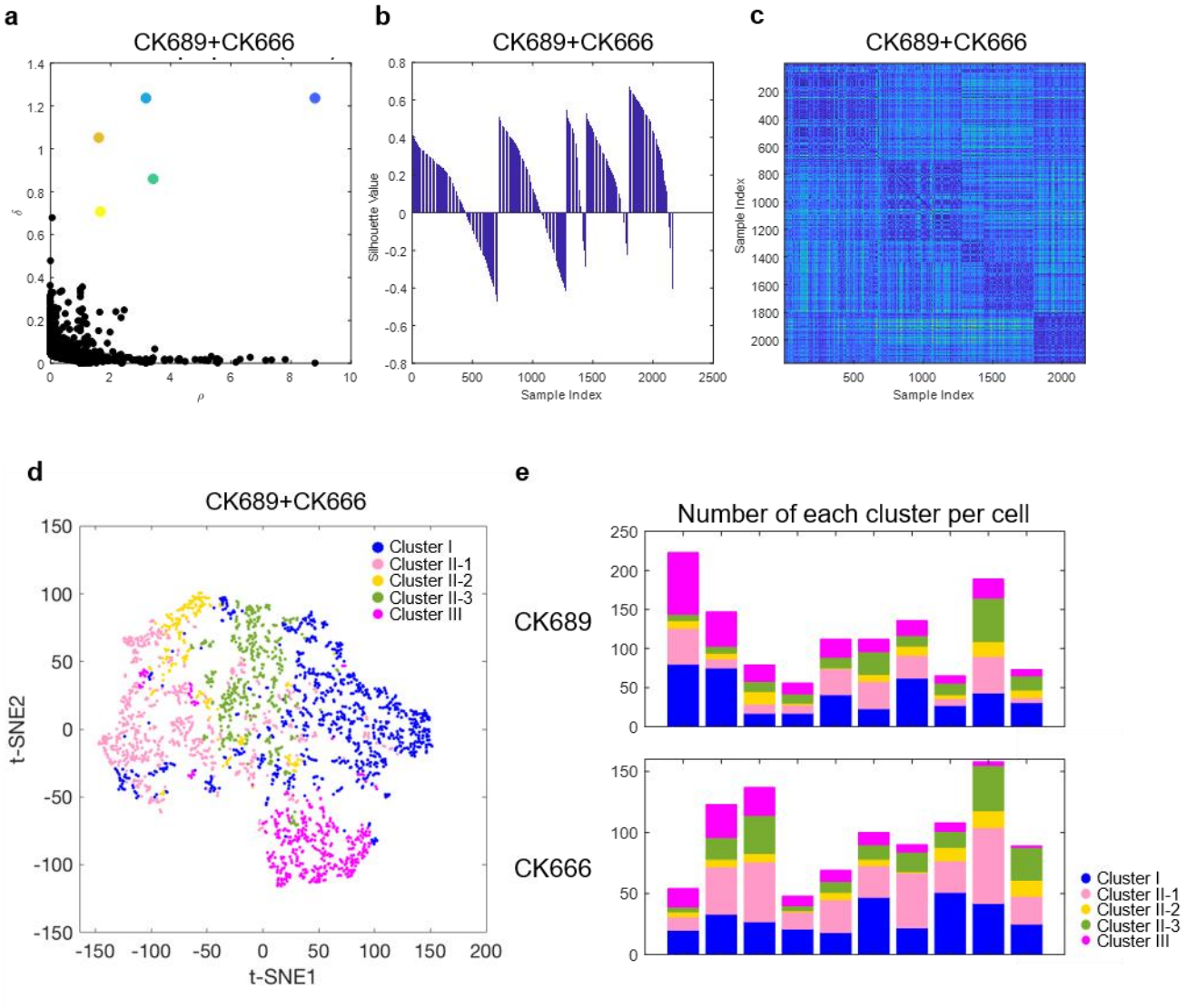
Supplementary Figure 3.3. Validation of clustering results and sample distribution of clusters (**a**) Davies-Bouldin Index, (**b**) Average silhouette, (**c**) Calinski-Harabasz pseudo F-statistic as a function of the number of clusters in density peak clustering. **(d)** MDS plot, **(e)** Silhouette plot, and **(f)** pair-wise order distance map for the validation of the clustering results. **(g)** Entire raw velocity maps for Cluster I, II-1, II-2, II-3, and III. **(h)** Sample distribution of clusters for individual cells.



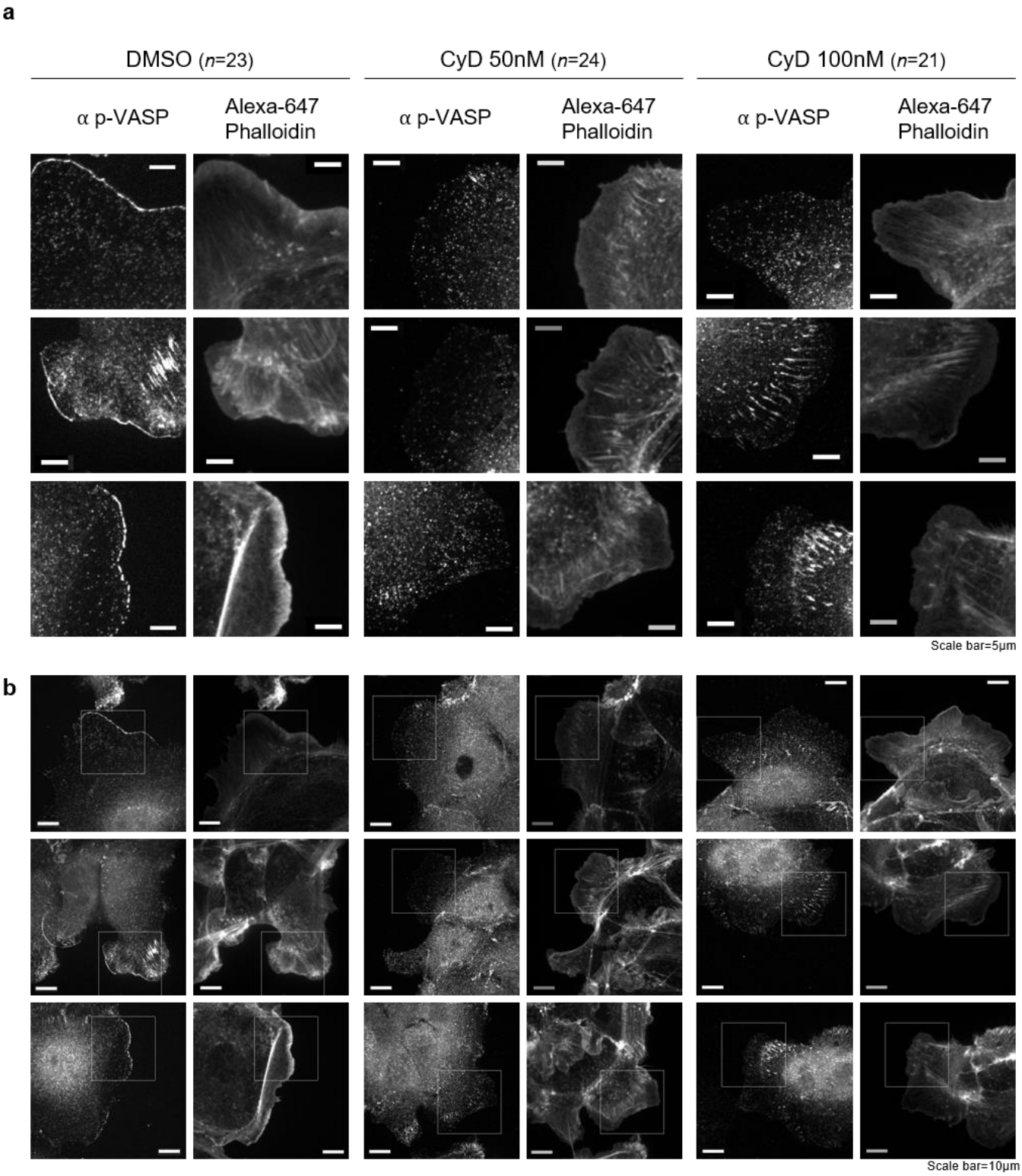
Supplementary Figure 3.4. Subcellular protrusion phenotypes of MCF10A cells (a) Decision graph of the Density Peak clustering analysis of protrusion velocities, **(b)** Silhouette plot, **(c)** pair-wise distance map, and **(d)** t-SNE plot for the validation of the clustering results. **(e)** Sample distribution of clusters for individual cells. **(f)** Averaged time series of protrusion velocity registered at protrusion onsets (t=0) in each cluster. Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals computed by bootstrap sampling. *n* is the number of sampled time series. Images from 16 cells were used for the analysis.



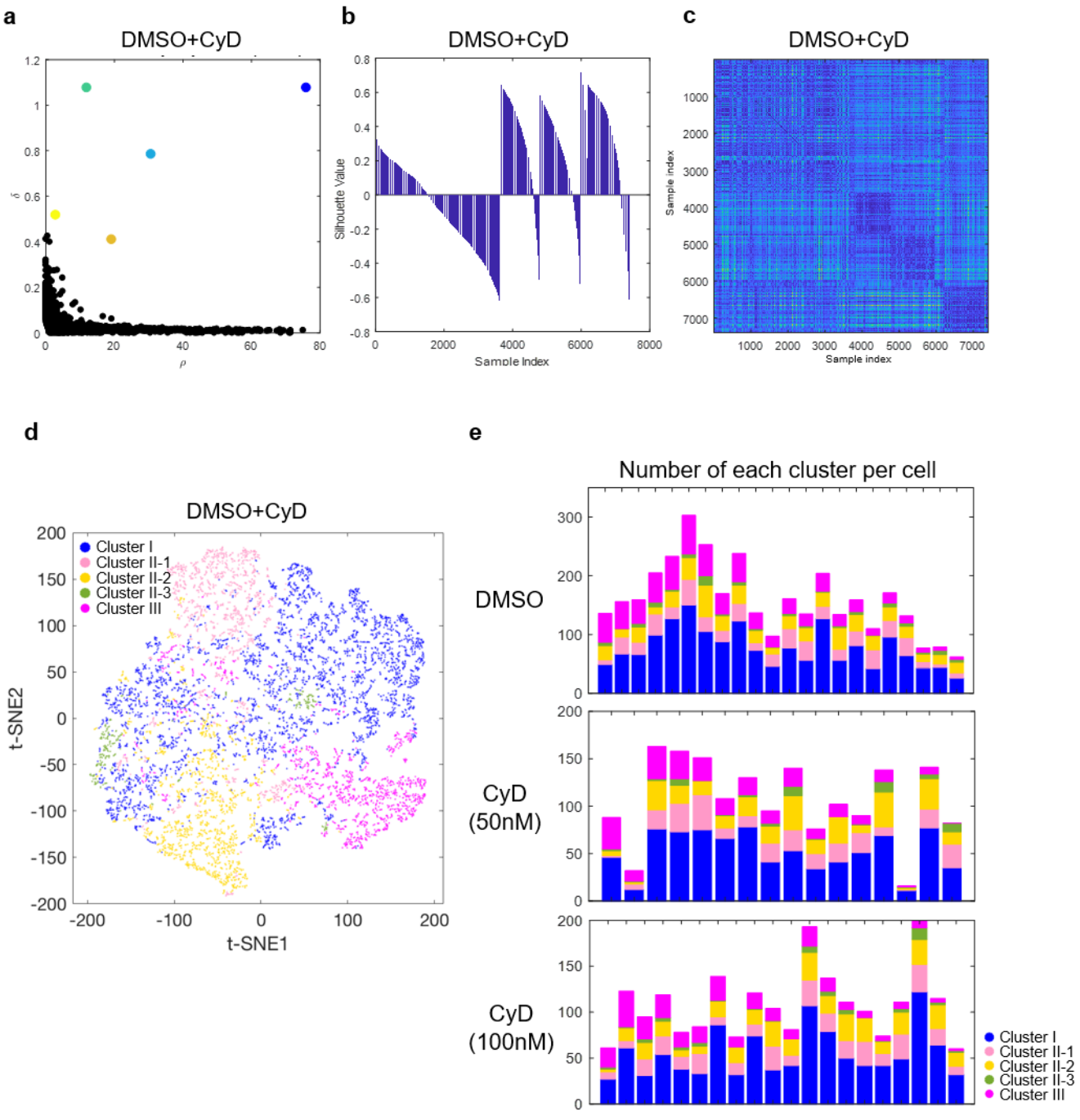
Supplementary Figure 3.5. Correlation analyses between protrusion velocity and actin regulator dynamics **(a)** Pairwise Pearson's correlation coefficients of protrusion velocity and HaloTag intensity time series registered relative to protrusion onset. The regions surrounded by the black lines are statistically significant correlation by Benjamini-Hochberg multiple hypothesis testing. **(b)** $-\log_{10}$(adjusted p-value) of pairwise Pearson's correlation coefficients of protrusion velocity and fluorescence intensity time series. **(c)** Scatter plots and Pearson's correlation coefficients of early Arp3 intensities and late protrusion velocities in each cluster. The permutation t test was used to calculate the p-values.



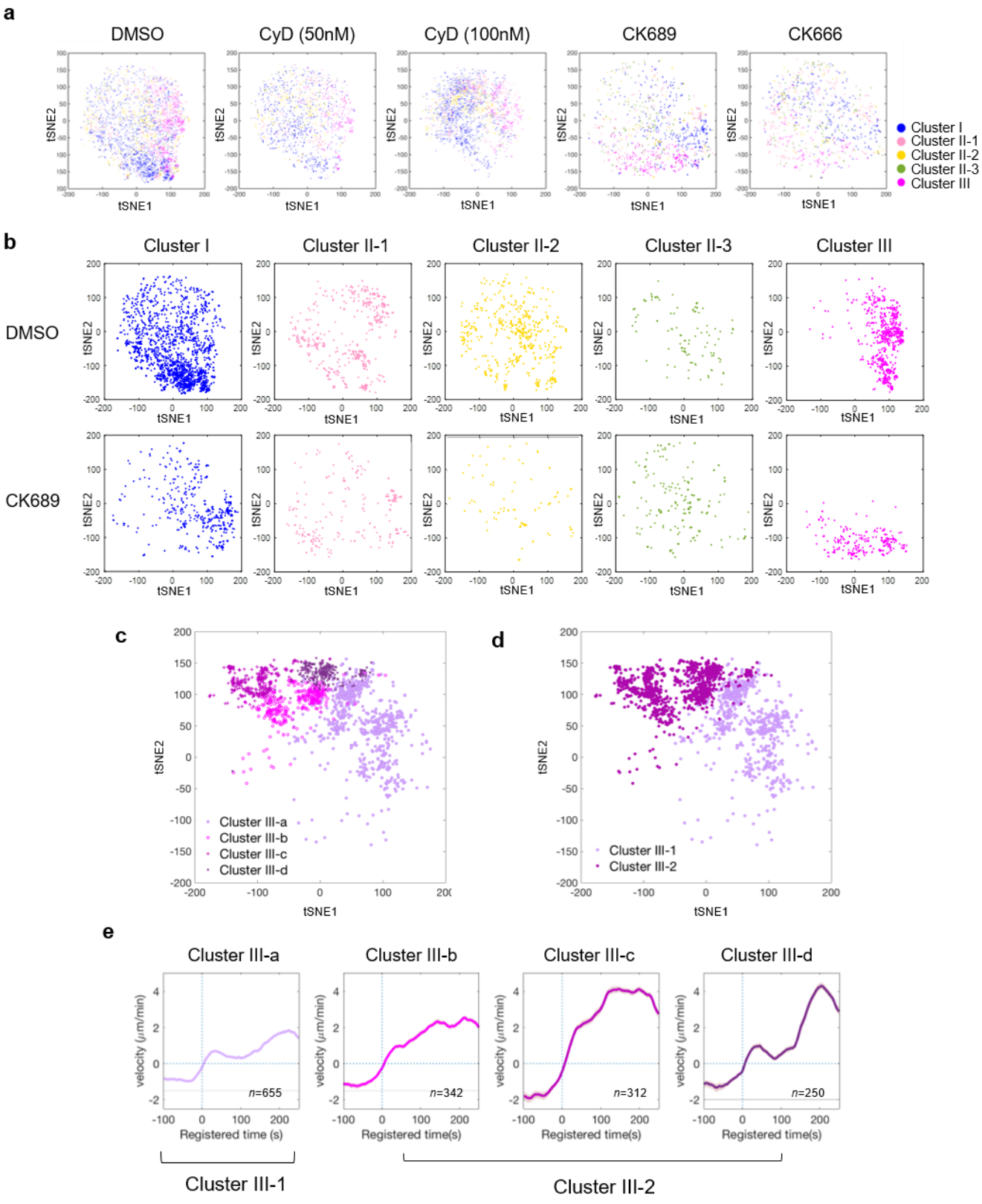
Supplementary Figure 3.6. Subcellular protrusion phenotypes of PtK1 cells treated with CK689 and CK666 **(a)** Decision graph of the Density Peak clustering analysis of protrusion velocities, **(b)** Silhouette plot, **(c)** pair-wise distance map, and **(d)** t-SNE plot for the clustering results. **(e)** Sample distribution of the clusters for individual cells treated with CK689/CK666.



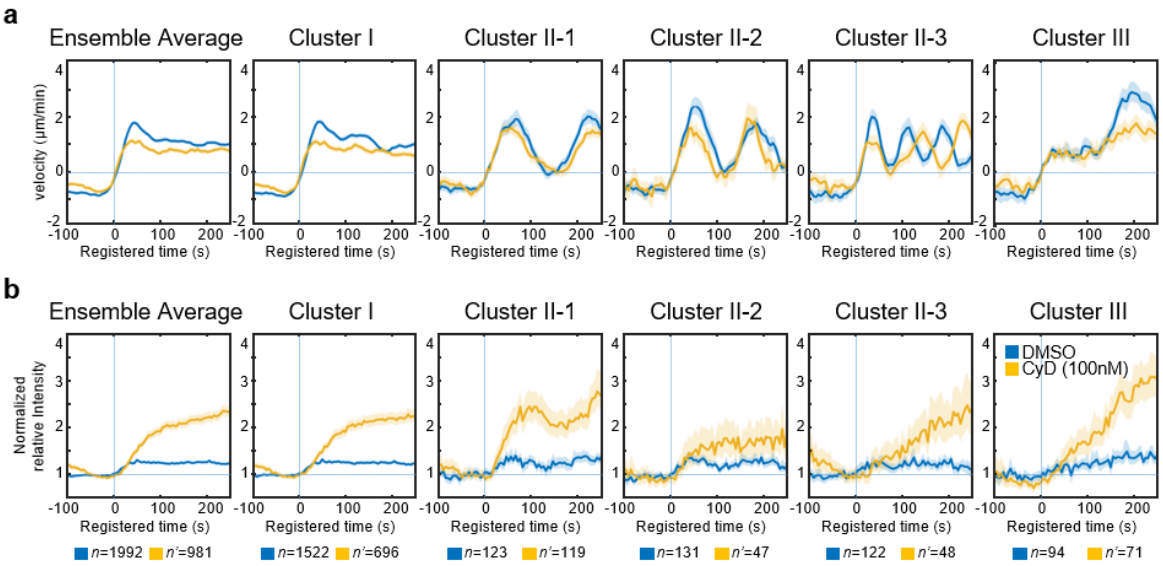
Supplementary Figure 3.7 Validation of effects of Cytochalasin D on VASP in the lamellipodia of PtK1 cells (a-b) Immunofluorescence images of PtK1 cells stained with anti-p-VASP and Phalloidin with and without Cytochalasin D treatment. (a) are the magnified images from (b). The number of images taken for DMSO-, CyD (50nM)- and CyD (100nM)-treated cells are 23, 24 and 21 respectively. Representative 3 images are selected and presented.



Supplementary Figure 3.8. Subcellular protrusion phenotypes of PtK1 cells treated with Cytochalasin D (a) Decision graph of the Density Peak clustering analysis of protrusion velocities, **(b)** Silhouette plot, **(c)** pair-wise distance map, and **(d)** t-SNE plot for the clustering result. **(e)** Sample distribution of the clusters for individual cells treated with DMSO and Cytochalasin D.


a
DMSO
CyD (50nM)
CyD (100nM)
CK689
CK666
tSNE1
tSNE2
Cluster I
Cluster II-1
Cluster II-2
Cluster II-3
Cluster III
b
Cluster I
Cluster II-1
Cluster II-2
Cluster II-3
Cluster III
DMSO
CK689
c
Cluster III-a
Cluster III-b
Cluster III-c
Cluster III-d
d
Cluster III-1
Cluster III-2
e
Cluster III-a
Cluster III-b
Cluster III-c
Cluster III-d
velocity (μm/min)
Registered time (s)
Registered time(s)
n=655
n=342
n=312
n=250
Cluster III-1
Cluster III-2

Supplementary Figure 3.9. Sub-clustering of accelerating protrusion (a) t-SNE plots of denoised protrusion velocity time series overlaid with the cluster assignments. **(b)** t-SNE plots of denoised protrusion velocity time series of individual clusters in control (DMSO/CK689) cells. **(c)** Sub-clustering of Cluster III by community detection. **(d)** Final sub-clustering by merging Cluster III-b/c/d to Cluster III-2. **(e)** Average time series of protrusion velocity registered at protrusion onsets (t=0) in each sub-clusters before merging. Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals computed by bootstrap sampling.



Supplementary Figure 3.10. Changes in Arp3 recruitment upon Cytochalasin D treatment (a) Ensemble averaged velocity time series of entire samples and averaged velocity time series of each cluster upon DMSO or Cytochalasin D (100nM) treatment in Arp3-expressing cells. **(b)** Ensemble averaged normalized fluorescence intensity series of entire samples and averaged normalized fluorescence intensity time series in each cluster upon DMSO or Cytochalasin D treatment in Arp3-expressing cells. All time series are registered with respect to protrusion onset (t=0). Solid lines indicate population averages. Shaded error bands indicate 95% confidence intervals computed by bootstrap sampling. (*n*: the number of time-series used for the analysis for DMSO treated GFP-Arp3 expressing cells, *n'*: the number of time-series used for the analysis for CyD treated GFP-Arp3 expressing cells. The numbers of the cells imaged for DMSO and CyD treatment are 15 and 11 respectively.