Electronic Supplementary Material (ESI) for Lab on a Chip.

SUPPLEMENTARY INFORMATION

Machine learning-aided quantification of antibody-based cancer immunotherapy by Natural Killer cells in microfluidic droplets

Saheli Sarkar, Wenjing Kang, Songyao Jiang, Kunpeng Li, Somak Ray, Ed Luther, Alexander R. Ivanov, Yun Fu, Tania Konry

SUPPLEMENTARY METHODS

Microfluidic device fabrication and droplet generation

The design of the integrated droplet generation and docking plaform was transferred from a transparency photomask (Fine Line Imaging, Colorado Springs, CO) to clean silicon wafers using UV photolithography. Briefly, the negative photo resist SU-8 2100 (MicroChem, Newton, MA) was deposited onto silicon wafers by spin-coating, baked and exposed to UV light (365nm) to obtain features of 150 μm height. The microfluidic devices were fabricated by standard soft-lithography protocols. A mixture of poly(dimethylsiloxane) (PDMS) preploymer and curing agent (10:1 ratio w/w) (Sylgard 184, Dow Corning, Midland, MI) was deposited evenly over the wafer. The mixture was degassed and cured for 12 hours at 65°C. The hardened PDMS layer containing the droplet network was removed from the wafer and bonded onto microscope slides that had been pretreated with plasma oxidation for 30-60 sec. The slides were heated at 90^{o}C for 10 minutes to complete bonding.

The device was treated with Aquapel glass treatment solution (Aquapel, Pittsburg, USA) for 15 minutes and air dried immediately before experiments. Cell suspensions and oil supplemented with 2% w/w surfactant (008-FluoroSurfactant, Ran Biotechnologies, Beverly, MA) were added to specifically designated inlets in the device through individual syringes. The oil used was Fluorinert® FC-40 (Sigma, St. Louis, MO). The flow of liquid suspensions were regulated by programmable syringe pumps (Harvard Apparatus, USA). The oil to aqueous flow rates were generally maintained at a ratio of 4:1 to obtain optimal droplet sizes.

Cell viability studies

Viability of cells in droplets was determined by Live/Dead Viability/Cytotoxicity assay (Life Technologies, Carlsbad, CA). Cells were suspended in media containing Calcein AM (2 µM final concentration) and ethidium homodimer-1 (4 µM final concentration) and loaded in droplets. Calcein AM was taken up by live cells and detected using excitation/emission wavelength of 494/517 nm. Ethidium homodimer labels dead cells and was measured at 528/617 nm. The ratio of total number of live cells to the total number of cells at any given time was reported as 'percentage viability'.

For co-encapsulation studies, target cells (SKOV3, SKBR3 etc) were labeled off-chip with Calcein AM for 30 minutes at 37°C. The labeled cells were washed twice to remove excess cell trackers and incubated with antibody solutions if required. The NK cells were left unlabeled. The two cell suspensions were loaded in separate syringes at an initial concentration of 2 million/mL for co-encapsulation studies.

Laser scanning cytometry

NK-92 cells (5×10^4/well) were labeled with a panel of antibodies from Human Cell Surface Marker Screening Panel (BD Biosciences, San Jose, CA) as per manufacturer's recommendation. Briefly, cells were suspended in BD Pharmingen Stain Buffer (FBS) supplemented with 5 mM EDTA, incubated with each primary antibody on ice for 60 minutes. Cells were washed three times and incubated with AlexaFluor® 647 conjugated goat anti-mouse Ig secondary antibody for 30 minutes in the dark. The cells were then fixed with 4% paraformaldehyde and counterstained with Hoechst 33342 (10 µg/mL) to label cell nuclei. The labeled cells were transferred to optical bottom black 96-well plates and analyzed using the iCyte® laser scanning cytometer (CompuCyte Corp. Westwood, MA). Hoechst stain was detected with 405 nm diode with 440/30 nm bandpass filter. Alexa Fluor 647 was detected with 633nm diode and 650 nm longpass filter. A 40X objective lens was used at spatial resoltion of 0.25nm. Events were segmented based on the Hoechst images and a panel of features were obtained for each cell, including the amount of DNA (Hoechst fluorescence) and the total amount of antibody fluorescence/cell. Data analyses were performed using the iCyte software (Version 3.4). For each well, the number of fluorescent cells was quantified and normalized to negative controls (cells labeled only with anti-mouse secondary antibody). Only data for the positive wells is reported.

Analyzing Cells via Deep Convolutional Neural Networks

1 Problem Definition

Given videos capturing the activity of the cell, we first focus on each frame of these videos. Each frame I consists of a pair of images captured at the same time, which includes a brightfield image I_1 and a fluorescent image I_2 . Given I as input, we aim to train two models to predict whether there is any interaction or death phenomenon in a single frame of the video. Here, we define the label L of a single frame I as follows:

$$L = [l_1, l_2] \tag{1}$$

where $l_1 \in [0, 1]$ and $l_2 \in [0, 1]$ are binary variables representing "interaction between immune and cancer cells" and "death of cancer cells" respectively.

2 Data Preprocessing

2.1 Channel Combination and Normalization

We have two channels of raw images, i.e., a brightfield image I_1 and a fluorescent image I_2. Both of I_1 and I_2 are linear single-channel gray-scale images. From our observation, both the brightfield and fluorescent images contain useful information for death and contact detection tasks. So we combine these two channels to obtain a multi-channel RGB image to train our convolutional neural networks. We repeat the normal image channel-wise for three times to obtain a RGB gray-scale image, and add the fluorescent channel to the Green channel.

$$I_R = I_B = I_1, I_G = I_1 + I_2 \tag{2}$$

$$I = \{I_R, I_G, I_B\}, \tag{3}$$

For each image sequence, we normalize the first RGB image to $[0, 1]$ by its maximum and minimum value of the first image of the image sequence. All following images after the first image are normalized in the same way, which ensures that the relative intensity is preserved for each image sequence.

$$I_{RGB} = \frac{I - \min(I)}{\max(I) - \min(I)} \tag{4}$$

The last step is to apply a Gamma correction to the linear RGB image to form an image in sRGB color space following the equation

$$I_{sRGB} = \{I_{RGB}\}^{\gamma} \tag{5}$$

where $\gamma = 2.2$. This operation simulates the visual process of human eyes, which enhances the contrast of the image.

2.2 Droplet Detection and Extraction

Each image Field of View (at 20X magnification) contains six droplets. We apply a circle detection algorithm which automatically detects circles around fixed length of diameters to detect the position of all the droplets in the image. We crop and resize each droplet to 200×200, and organize the images of a single droplet into a time-series image sequence.

For the label of each image, based on the format defined in Eq. 1, we assign more specific death phenomenon of cells. When death occurs in droplets, we set $l_1 = 1$, otherwise $l_1 = 0$. Similarly, for contact analysis, as long as there is contact between the two cells, we set $l_2 = 1$, otherwise $l_2 = 0$. Based on this definition, we pre-process the original label excel file to obtain the labels for each frame in our format (saved in .npy files).

2.3 Contrast Enhancement Using Histogram Equalization

To amplify the signals contained in the images, higher local and global contrast is desired. We apply an Contrast-limited adaptive histogram equalization (CLAHE) algorithm [5] to further enhance the images in the acquired dataset. CLAHE is capable of increasing both the contrast adapted to the local regions, while preventing noise from being amplified. The above figure shows an example of CLAHE result, which reveals details that are previously hidden in the image.

2.4 Image Augmentation

The cell image dataset we acquire is relatively small in size compared to some commonly used machine learning datasets such as ImageNet[1] and COCO[2]. Deep learning models always require a large dataset to learn its parameters. Training on a small dataset will easily overfit the network and harm the performance. To overcome this drawback, we apply an image augmentation technique on the training images to increase the diversity and scale of the acquired dataset. Image augmentation can also improve the overall performance and increase robustness of the trained model.

2.4.1 Random Flip

We randomly flip the input image with probability of 0.5 to obtain its mirror image. This operation approximately doubles the size of the acquired dataset.

2.4.2 Random Crop

For the training images, all images are resized to 256×256 and randomly cropped to 224×224. This process simulates the pixels and position shifting of the images during image capturing process. For the testing images, we resize the images to 256×256 and crop them to 224×224 in the center.

3 Convolutional Neural Network

3.1 Model Structure and Configuration

Since we formulate the death prediction and contact prediction tasks into classification tasks, we build two deep learning models to handle these tasks. We use VGG network [4] to extract meaningful features from the input images followed by a softmax classifier to output the classification results, as illustrated in Figure 1 of the manuscript.

3.1.1 VGG based feature extraction

We use VGG-19 as our backbone network structure and build two models, each focusing on binary classification, of death and interaction respectively. VGG network is a commonly used deep learning network designed for classification tasks, which contains a number of convolutional layers followed by activation layers and max pooling layers. Convolutional layers perform as a group of trainable filters. We use Rectified Linear Unit (ReLU) function as our activation layers. ReLU is a simple non-linear function, as shown in Equation 6.

$$f(x) = \max(0, x). \tag{6}$$

ReLU tends to make the neural network sparse and speed up the training process. Another important feature is that ReLU layer make sure that each convolutional layer is performing a non-linear operation, which is very important to the convolutional neural network. MaxPooling layers downsample and reduce the volume of the intermediate extracted features. In our VGG based networks these layers are configured in a hierarchical way, which extracts features from low-level to high-level. Detailed architecture of the network can be found in Table 3. The final output features are input to the classifier to do classification prediction.

3.1.2 Death and Interaction Classifiers

For both the networks of death and interaction prediction, we use three layers of fully-connected neural networks followed by a Softmax activation function to build the classifiers. The fully-connected layers take the extracted features as input to predict the scores of the input image belonging to the corresponding classes. Then, the Softmax activation function normalize these scores to their normalized probabilities, which sum up to one. The probabilities are used to calculate classification loss to optimize the network.

3.1.3 Cross-entropy Loss Function

Both the death prediction and interaction prediction can be formulated as binary classification tasks. Therefore, the objective function to train the whole network can be described as a cross-entropy loss,

$$\mathcal{L} = -(y \log(p) + (1 - y) \log(1 - p)), \tag{7}$$

where $\log$ is the natural $\log$ function, y is a binary indicator (0 or 1) if the class label c is the correction classification, and p is the predicted probability of the input observation x is of class c. The optimizer runs back propagation using $\mathcal{L}$ to update the parameters of our neural networks.

3.2 Pretraining and Knowledge Transfer

A deep learning network such as VGG network normally contains a huge amount of weights from its convolutional layers. Training such amount of weights often requires a large-scale dataset to avoid overfit and improve performance. Our acquired dataset is considered a small-scale dataset in machine learning aspect. To solve this problem, we pretrain the VGG network on a commonly used large-scale classification dataset called ImageNet [1]. The pretraining process transfers the knowledge from the large-scale dataset to our classification tasks. Specifically, it can help our network to analyze its low-level and high-level filters to recognize common patterns such as texture, shape and colors. After pretraining the network on ImageNet, we adjust the network and train the classifiers using the acquired cell images.

3.3 Considering Temporal Information

For predicting death of the cancer cells, we further design a framework that takes temporal information into consideration. We use the supposition that once a cancer cell is confirmed as 'dead' at the i-th frame of video, it will not be alive in the following $k > i$ frames of the same video.

Therefore, we apply a sliding window of size 5 on the video frames; if 3 or more frames in the sliding window classify the cell as 'dead', we confirm definite death of the cell and all subsequent frames are listed as 'dead'.

4 Experiments

4.1 Implementation Details

We use Pytorch [3] to build and train our models. Pretrained model of VGG19 is loaded for finetuning. We set the batch size to 20 and use SGD with learning rate of 0.001 to train the two models. We split the dataset into training set and test set for cross-validation. We validate the models on the validation set every epoch and save our best models. The total epochs is set to 80.

4.2 Results of Death and Interaction Prediction

In order to verify the effectiveness of our models, we do experiments on the data after the pre-processing. We randomly divide the whole dataset into two parts, 73 videos with 5865 images for training and 21 video with 1425 images for validation. After the training, we find our models could achieve overall accuracy of 96.77% on death prediction and 93.54% on interaction prediction, as shown in Fig 1B and 1C.

4.3 Ablation Study on Data Preprocessing

We conduct an ablation study on the effectiveness of our data preprocessing techniques including the combination and normalization of brightfield channel and fluorescent channel images, CLAHE contrast enhancement and data augmentation. As shown in Table 1, each component of our data preprocessing techniques contribute to our final prediction accuracy.

Models	Death Acc. %	Interact Acc. %
I_1 only	89.87	80.48
I_2 only	91.56	75.19
I_1 + I_2	92.35	83.27
+ CLAHE	95.93	83.75
+ Data Augmentation	96.77	93.54

Table 1: Ablation studies of prediction accuracy on data preprocessing including multi-channel combination, CLAHE contrast enhancement and data augmentation.

4.4 Death Prediction Model with Temporal Information

We compare the accuracy of death prediction with and without temporal information to show the effectiveness of our death prediction model with prior temporal knowledge. As shown in Table 2, the death prediction accuracy can be improved by around 3 percent with a sliding window of size 5. Adding the temporal factor allows us to achieve better performance accuracy for the time-course of a cell than focusing on a single image in the sequence..

Models	Accuracy %
w/o Temporal Model	93.75
w/ Temporal Model	96.77

Table 2: Temporal Model represents the death prediction model considering temporal information and priors knowledge. The length of the sliding window is set to 5.

Layer	Kernal Size	Filter Size	Stride	Padding
Input $3 \times 224 \times 224$				
conv	3×3	64	1	1
conv	3×3	64	1	1
maxpooling				
conv	3×3	128	1	1
conv	3×3	128	1	1
maxpooling				
conv	3×3	256	1	1
conv	3×3	256	1	1
conv	3×3	256	1	1
conv	3×3	256	1	1
maxpooling				
conv	3×3	512	1	1
conv	3×3	512	1	1
conv	3×3	512	1	1
conv	3×3	512	1	1
maxpooling				
conv	3×3	512	1	1
conv	3×3	512	1	1
conv	3×3	512	1	1
conv	3×3	512	1	1
maxpooling				
FC-4096				
FC-4096				
FC-2				
softmax				

Table 3: Network architecture. We follow a VGG-19 architecture for both our death and interaction prediction model. Abbreviations: conv: convolutional neural network with rectified linear unit (ReLU) activation, FC: fully connected neural network.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009.

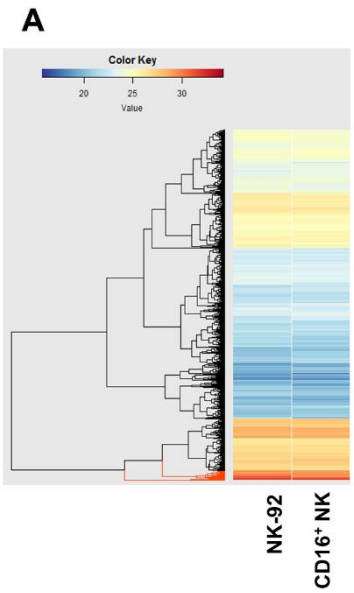
[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*. Springer, 2014.

[3] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[4] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] K. Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics gems IV*, pages 474–485. Academic Press Professional, Inc., 1994.

SUPPLEMENTARY FIGURES



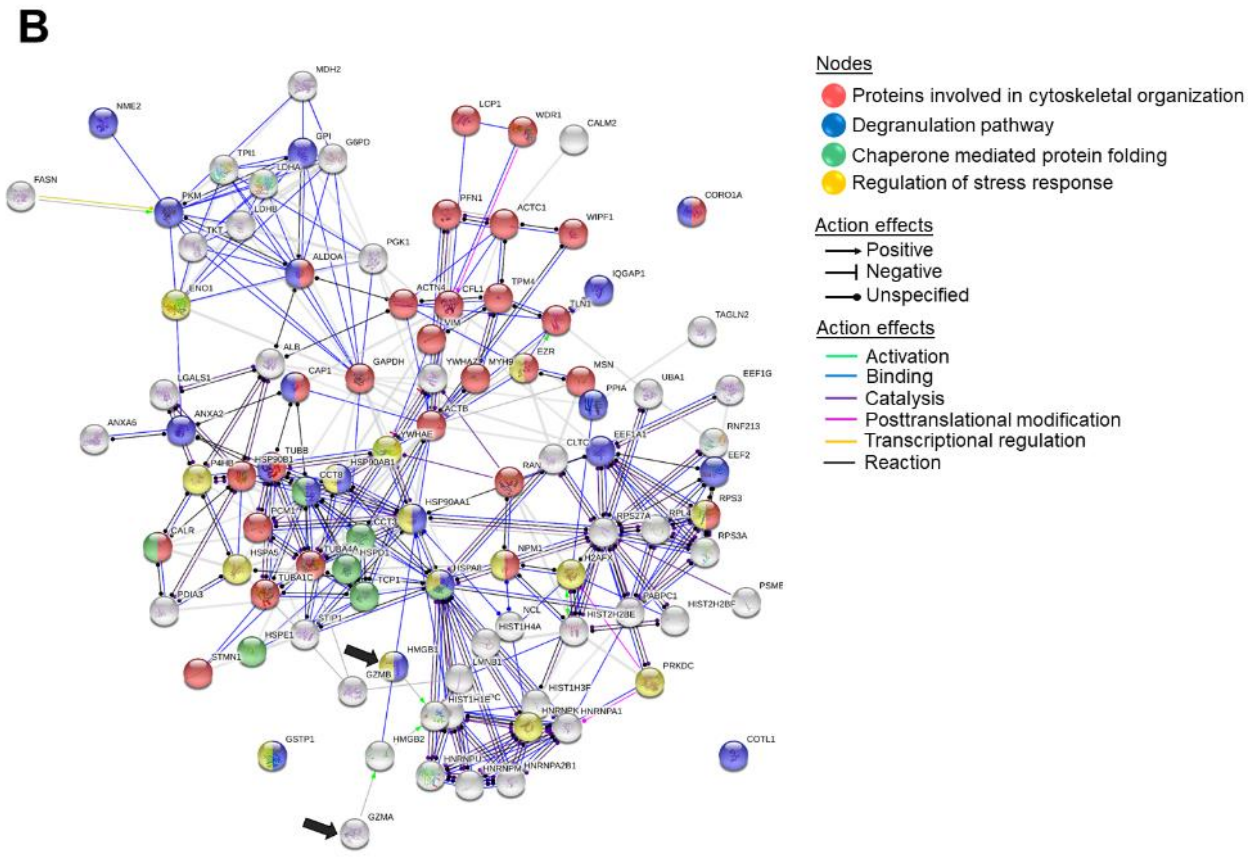


Fig. S1. (A) Clustered heatmap of average log2-intensities of proteins quantified in at least 50% of the replicates for each cell type. Based on hierarchical clustering, the most abundant protein cluster is indicated in red. (B) Network connections for proteins that were most abundant as annotated by Gene Ontology (GO) pathway enrichment analysis. Colored nodes are associated with specific pathways (i.e., first shell of interactors) and white nodes represent second shell of interactors. Red nodes: proteins involved in cytoskeletal organization. Blue nodes: degranulation pathway. Green nodes: Chaperone mediated protein folding. Yellow nodes: Regulation of cell response to stress. The biological processes are indicated by colored lines. Black arrows point to Granzymes A and B, two characteristic NK cell markers.

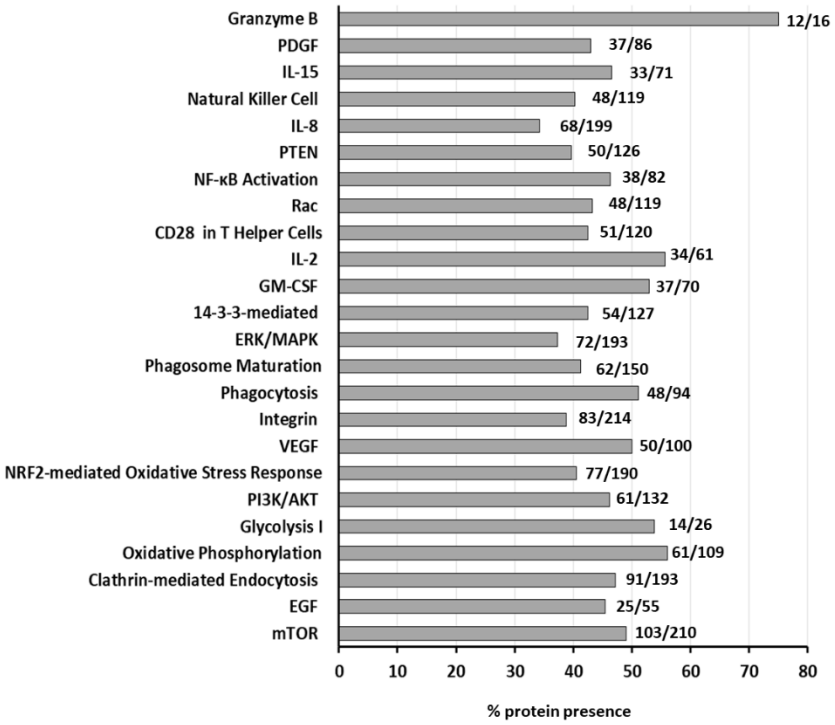


Fig. S2. Functional classifications of specific pathways detected significantly in parental NK-92 cells by Ingenuity pathway enrichment analysis. Significance of the canonical pathways was determined by the default threshold in IPA [-log(p value) >1.3]. The number of proteins involved in each pathway with respect to total number of proteins in the pathway are listed.