Classifying T Cell Activity with Convolutional Neural Networks

				Q												0
9 ⁻⁰⁰¹		00 00	o 7iii	e (Ja	av) V	Vand	8 A	Anto	nv (Gitte		0.0				
					A	oril 1	3, 2	019			0			Q	0_	
				6						8					0	
		9÷			0								6			

Why Classifying T-cell?



CAR T-cell Therapy

Image Data



Activated







CellProfiler



4,463 images, 6 donors



8,579 images, 6 donors

Image Processing: Image Size Matters







Donor 5: Image Size Comparison

Activated

Image Processing









Augment

Augment







Image Filtering

Classification Model Summary

Frequency Classifier	Pred
LR with intensity matrix	Logistic re
LR with total intensity + size	Logistic regr
LR with CellProfiler features	Logistic regre
Simple Fully Connected NN	Fully conr
LeNet	Simple cor
Retrained CNN (Last Layer)	Retrain the
Retrain CNN (Best Layers)	Retrain th

ict probability by the positive frequency in train donors

egression model with padded and resized image pixel matrix (82*82)

ression model with two numbers, total intensity and mask size for each image

ession model with 123 "intensity", "texture", and "area" related features.

nected one-layer neural network. Tuned on neuron number, learning rate and batch size

nvolutional neural network model, tuned on learning rate and batch size

e last layer of Inception v3 model, tuned on learning rate and batch size

ne last "n" layers, tuned on "n", learning rate and batch size

Transfer Learning



14 million images







10k images

1k categories

Transfer Weights



Skip Training

2 categories













ponor ponor 2 nor 3 nor 5 nor 6



Inner Loop of Nested Cross-Validation

Outer Loop



Inner Loop of Nested Cross-Validation

Outer Loop

Model Summary: Metrics





Sample Size

- Retrained CNN (Lat Layer)
- Retrained CNN (Best Layer)

1500



500

Activated Quiescent





Donor 5





Donor 5

Retraining More Layers = Better?

Retraining More Layers = Better?

Interpretation: Logistic Regression (Pixels)

Logistic Regression(Lasso) Coefficient

exp(coef)									
	1.02								
	1.01								
	1.00								
_	0.99								
	0.98								
	0.97								

Not used feature 1.00

Interpretation: Transfer Learning

UMAP Plot: Neural Network Features v.s. Raw Features

Transfer Learning: Saliency Map

Transfer Learning: Misclassified Images

Predict: Activated Softmax: 0.08, 0.92 Confidence: 0.28, 0.72

Predict: Activated Softmax: 0.13, 0.87 Confidence: 0.32, 0.68

Hold-out Donor 4: Did Not Break!

- Retrained CNN (Best Layer)

Sample Size

Reproducible Research

- "How did I make that plot?"
- "What images did I use to train model 1?"
- "Why my code stop working?"

- Organize files
- Save code and data
- Version control

Karl Broman, <u>http://kbroman.org/steps2rr/</u>

Reproducible Research

```
# AUC plot
p3 = ggplot() +
   scale y continuous(sec.axis = sec_axis(~ 200*., name="Sample Size"),
                      breaks = scales::pretty breaks(n = 10),
                      limits = c(0, 1)) +
   geom_col(data=count_df, aes(x=donor, y=count/200, fill=variable),
             position = position stack(reverse = TRUE)) +
   geom_line(data=metric_df, aes(x=donor, y=auc, color=method, group=method),
             size=line size, alpha=0.5) +
   geom point(data=metric_df, aes(x=donor, y=auc, color=method),
               size=point size, alpha=0.8, stroke=0) +
   scale fill manual(values=c(alpha("#A2A2A2", 0.5), alpha("#E3E3E3", 0.5))) +
   ggtitle("AUC") +
   labs(color="Model", fill="Activity", x="Donor",
        y="") +
   theme minimal()+
   theme(plot.title = element_text(hjust = 0.5, size=20),
          axis.title = element text(size = 20),
         legend.title = element text(size = 20),
          legend.text = element text(size = 20))
# There is a bug of calling ggarrange with common legend in R magic, we
# only plot legend for AUC and manully set up widths here
# ggarrange(p1, p2, p3, ncol=3, legend='right', common.legend=TRUE)
```

ggarrange(p1, p2, p3, ncol=3, widths=c(1,1,1.7))

https://github.com/gitter-lab/t-cell-classification

Take-away Image pre-processing is very important

microscopy images

- Cloud computing and distributed system
- Transfer learning does work on single-cell
- Reproducible research