

# Deep Generative Networks in Single Cell Genomics

Panagiotis Papasaikas

FMI Computational Biology

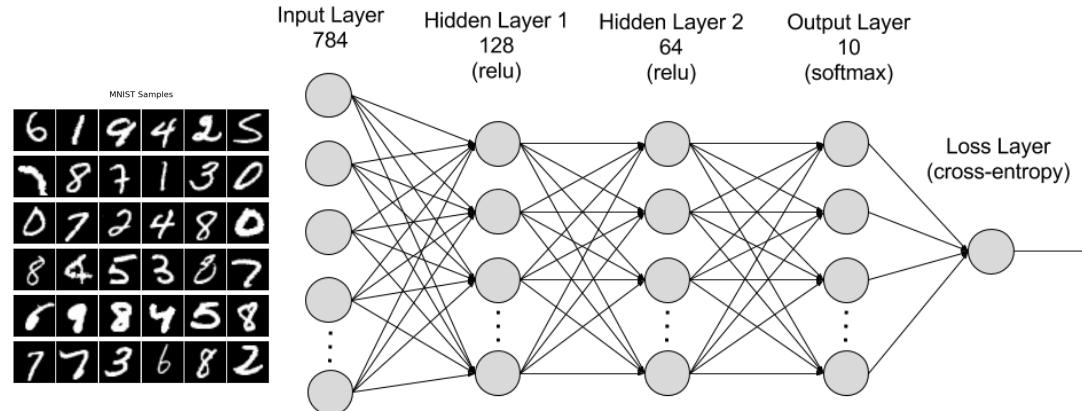
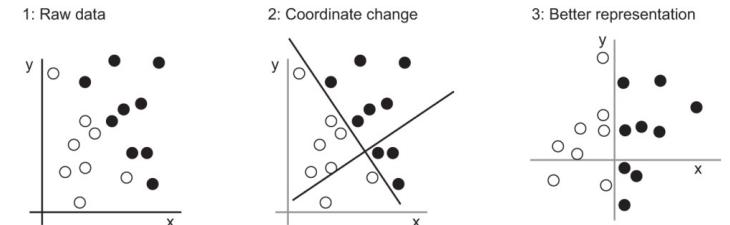
## Overview

- Introduction to Deep Learning\*
- High Level APIs for Deep Learning
- Representation Codes
- DGNs
  - VAEs
  - GANs
- Applications in scingle cell-omics and existing tools (non-comprehensive)
- Group project overview
- Perspectives

\*Parts of the introduction to DL inspired by J.J. Allaire's keynote at rstudio::conf 2018 and Franchoit Chollet's "**Deep Learning with R**"

# What is Deep Learning

Deep Learning Models take an input and transform it to an output via successive layers of increasingly abstract and meaningful **representations**



Raw data      Extraneous information  
                  filtered, useful information extracted

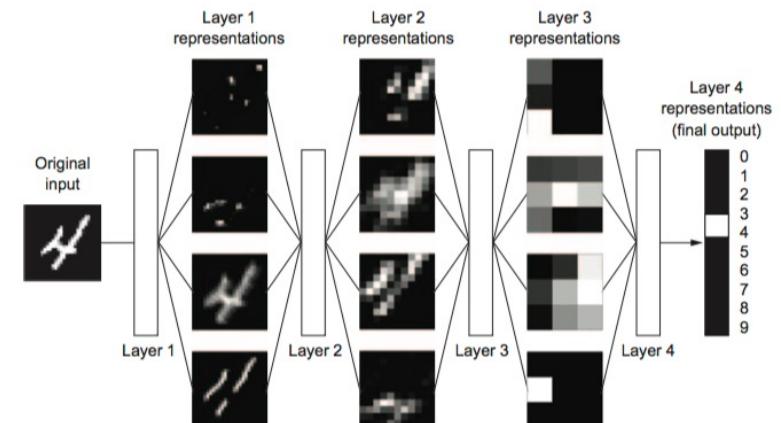


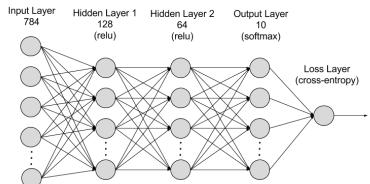
Figure 1.6 Deep representations learned by a digit-classification model

Image from F. Chollet's "Deep Learning with R"

!!! What is a “meaningful representation” is a relative concept that depends on the task at hand

Why Deep? -> Multi Layered Representation

# The mechanics of model training



The **loss function** measures the success of the model for the task at hand.

The parameters (weights) of the model are updated towards a direction that provides an improvement

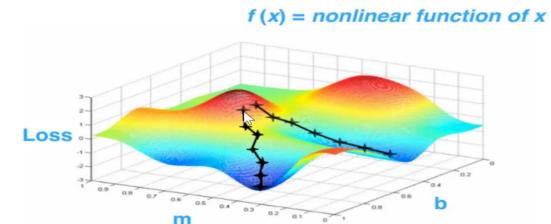
Updates are done using the **backpropagation** algorithm and the **chain rule** that traverses the model from the output towards the input

The direction towards which the parameters need to move is computed using **Stochastic Gradient Descent** variants

This loop is repeated many times using small splits of the data (batches)(epochs) until convergence

optimizer

## Gradient Descent



## What spurred the revolution?

Mainly advances on three fronts:

- **Massively parallel computation hardware** (GPUs, TPUs)
- **Improved algorithms**  
robust backprop, optimizers, regularization techniques
- **High-quality (often labeled) datasets**  
web usage, advances in tech/instrumentation in hard sciences



Improved architectures

User-friendly platforms

## Successes of Deep Learning

- Refined web-searching
- Spam/Fraud detection
- Near-human image classification (**MSRA, ImageNET**)
- Near-human machine translation (**DeepL**)
- Superhuman chess/GO playing (**AlphaZero, LC0**)
- Autonomous driving
- Natural language processing (e.g **IBM debater, GPT-x**)
- Protein Folding
- Medical Image Processing
- Drug design
- Diagnostics

# High level APIs for Deep Learning: Keras, TensorFlow and beyond.

Keras as a high level API supports multiple DL backends:

Keras API spec

TensorFlow

Theano

CNTK

Multiple Deep Learning frameworks:



## What is Tensorflow

- TF is am open source general purpose numerical computing library (not only DL, e.g general optimization libraries).



**TensorFlow**

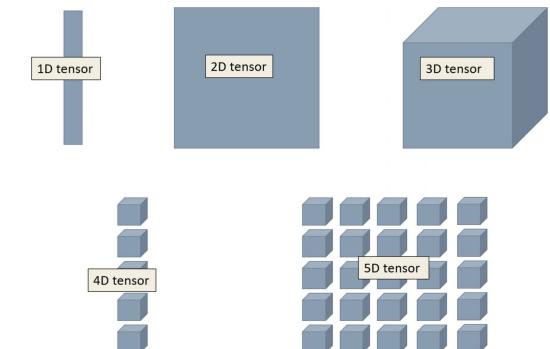
- Originally developed by engineers in the Google Brain Team for conducting ML research
- Hardware independent (CPUs, GPUs, TPUs)
- Supports large datasets/distributed execution

# The model building blocks in Tensorflow/Keras

- Tensors are multidimensional arrays.

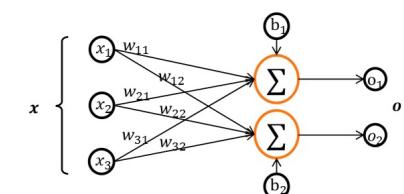
Data	Tensor dimension	R object
Cell label	1D (samples)	vector
Gene Count Matrix	2D (samples, genes)	matrix
Longitudinal data	3D (samples, genes, timestamp)	3d array
Microscopy Images	4D (samples, height, width, channels)	4d array
Video	5D (sample, height, width, channel, frame)	5d array

\*Notice the orientation convention is opposite to what bioinformaticians / R users are used to



- Layers are units of numerical computations (transformation functions) applied on tensors and **parameterized by weights**.

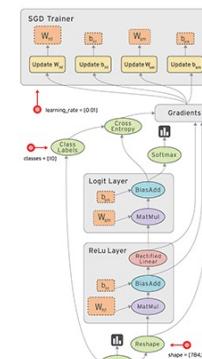
e.g addition, matrix multiplication, sampling, taking gradients...



- Layers and Tensors are combined to construct computation graphs (DAGs). Nodes are layers (computations), edges are Tensors.

Tensors “flow” through the computation graph and do smth useful (?).

A fully specified graph from input to output is a Model.



TensorFlow graph CC  
by [Tensorflow.org](https://Tensorflow.org)

# Keras



- Keras is a high level API that provides convenient wrappers for commonly used layers or computation graphs

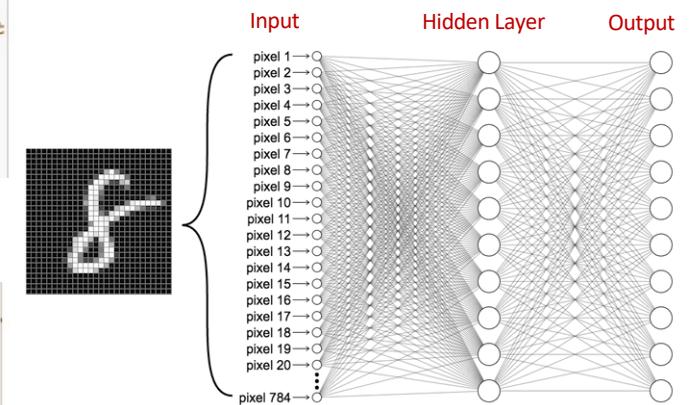
```
#defining a keras sequential model
model <- keras_model_sequential()

#defining the model with 1 input layer(784 neurons), 1 hidden layer(784 neurons) with dropout rate 0.4 and 1 output
#i.e digits from 0 to 9
model %>%
layer_dense(units = 784, input_shape = 784, activation = 'relu') %>%
layer_dropout(rate=0.4) %>%
layer_dense(units = 10,activation = 'softmax')
```

```
#defining model with one input layer[784 neurons], 1 hidden layer[784 neurons] with dropout rate 0.4 and 1 output 1.
model=Sequential()

from keras.layers import Dense
model.add(Dense(784, input_dim=784, activation='relu'))
keras.layers.core.Dropout(rate=0.4)

model.add(Dense(10, input_dim=784,activation='softmax'))
```



MLP model for digit classification



$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

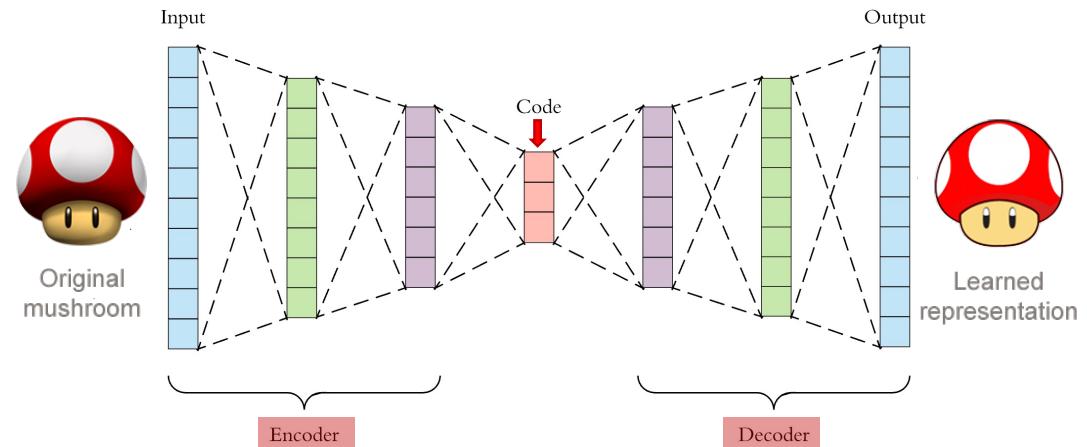
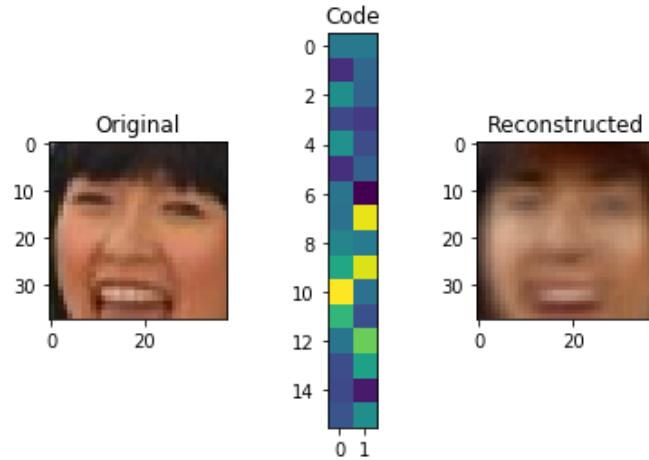
relu activation

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

softmax activation

## Autoencoders: architecture and latent codes

- Unsupervised (easy access to large training sets)
- Objective is to obtain an output that matches the input.
- Data are “squeezed” through successive layers of decreasing dimensions
- The middle hidden layer is a **code** (latent code) that **represents** the input:

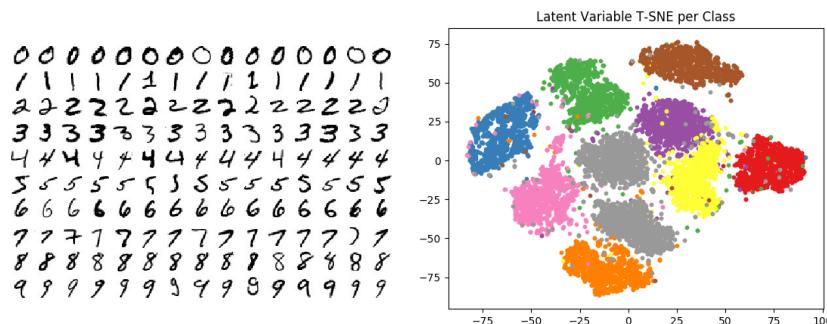


### Multiple AE flavors

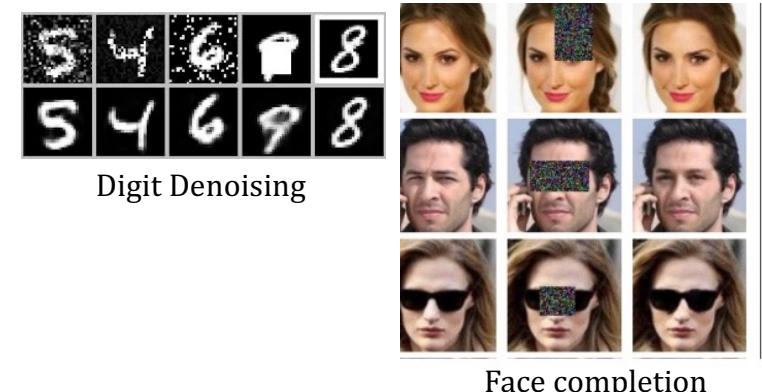
Deep/Stacked, Sparse, **Variational**, Denoising, Adversarial, Disentangled...

## Applications of AEs

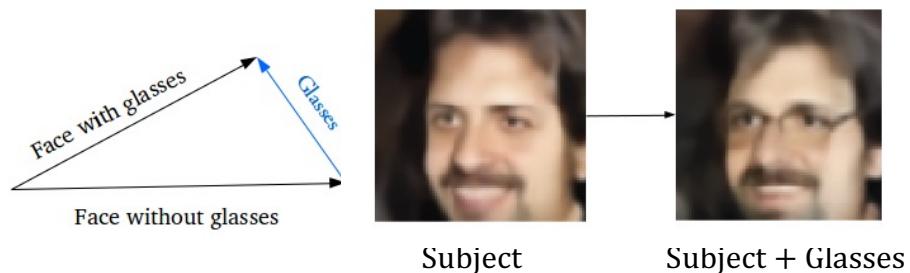
## 1. Dimensionality reduction & visualization



## 2. Denoising & completion (imputation)



### 3. Feature manipulation, interpolation and exploration



## Multiple AE flavors

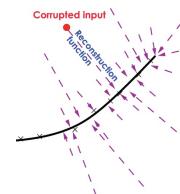
## Deep/Stacked, Sparse, Variational, Denoising, Adversarial, Disentangling...

## Why AEs for SC transcriptomics?

## AE extensions and “good” representation codes

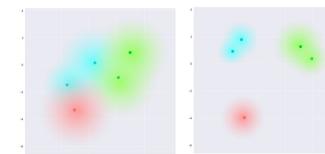
The common goal it to obtain a good code representation of the input data

- Robust to “meaningless” input corruptions

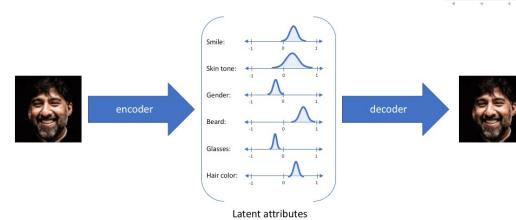


- Generalizable  $\Rightarrow$  can transfer to multiple settings /related problems

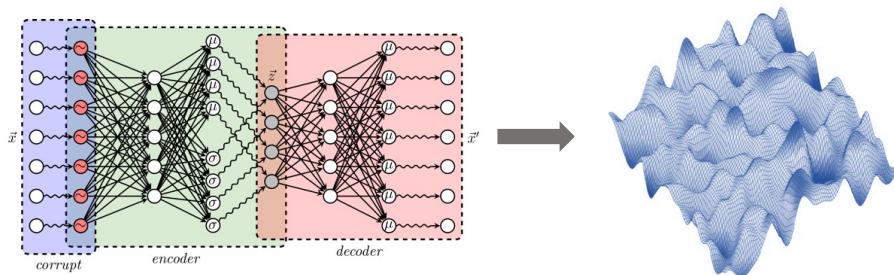
- Smooth / Coherent: similar inputs  $\mapsto$  similar codes.



- Explanatory

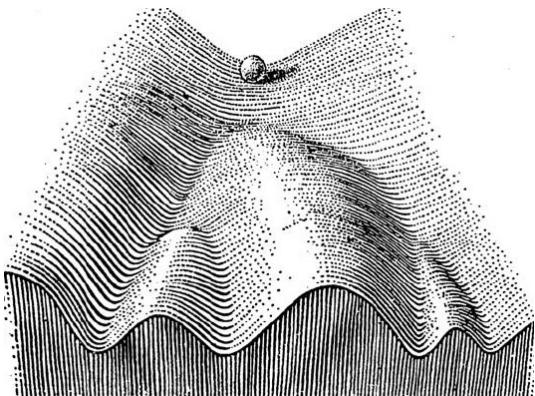


The latent representation is an estimation of the underlying manifold that gives rise to the data

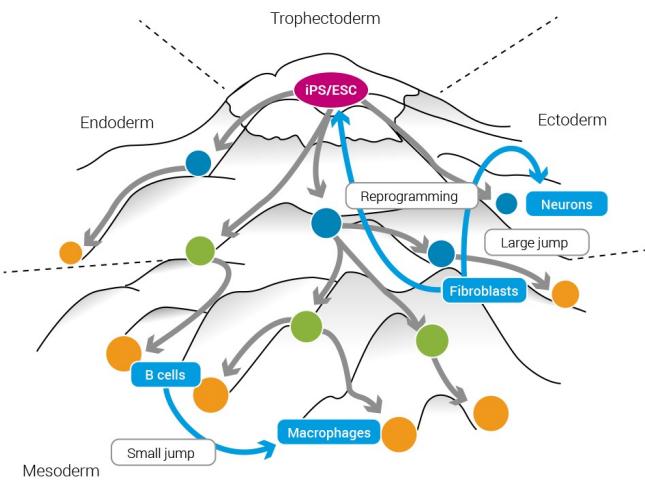


- Succinct, generative representations of complex Tx manifolds.
- Each location in this manifold represents a different realizable cell-state

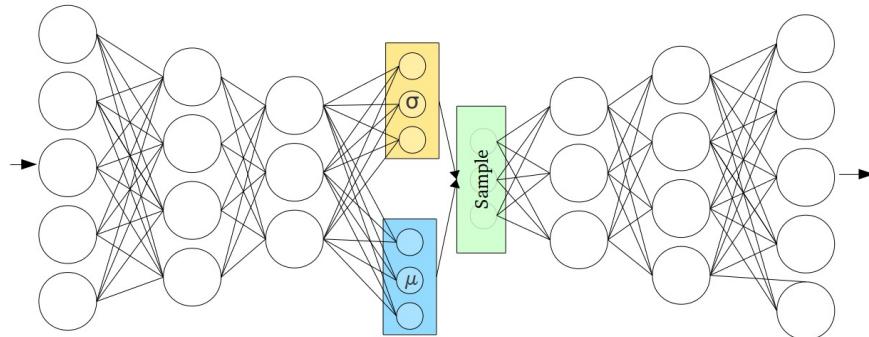
A useful analogy:



Waddington landscape (1956)



## Common architectures in SC-omics 1: Variational Autoencoders



- VAEs generalize AEs adding stochasticity
- Encourage a continuous latent manifold
- Robustness + valid decoding
- Allows interpolation and exploration

D. P. Kingma and M. Welling. "Auto-encoding variational Bayes". arXiv:1312.6114, 2013.

$$\mathcal{L}_\beta = \frac{1}{N} \sum_{n=1}^N (\mathbb{E}_q[\log p(x_n|z)] - \beta D_{KL}(q(z|x_n)||p(z)))$$

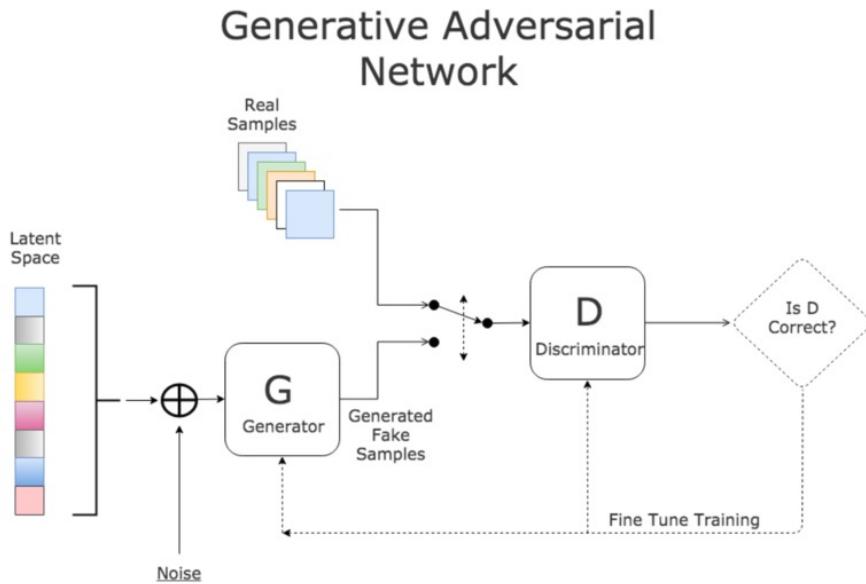

---

Reconstruction
Distance to latent prior

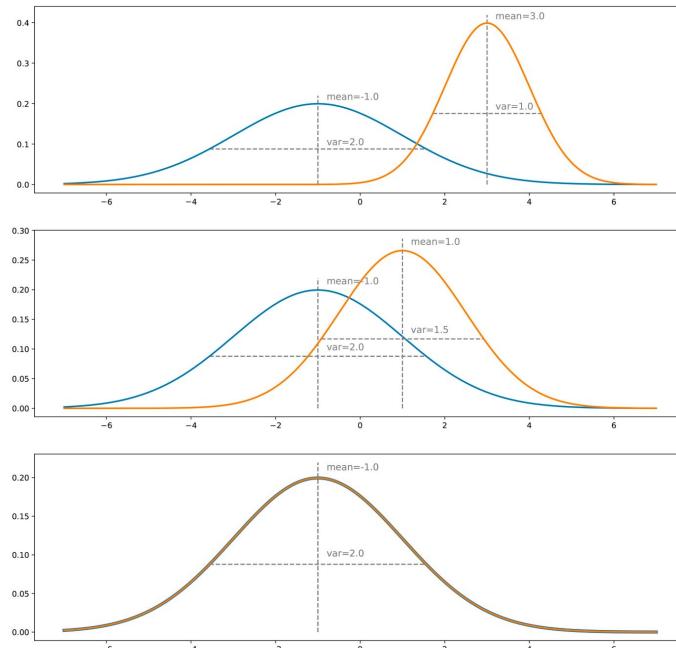
The latent prior is a multivariate normal with a unit covariance matrix

- $\beta = 1$  : ELBO (Evidence Lower Bound, standard VAE)
- $\beta < 1$  : Partially regularized VAE (Liang et al. 2018)
- $\beta > 1$  : Disentangling Autoencoders ( $\beta$ -VAE, Higgins et al. 2017)

## Common architectures in SC-omics 2: Generative Adversarial Networks (GANs)



I. Goodfellow, J.Pouget-Abadie, M.Mirza, B.Xu, D.Warde-Farley, S. Ozair, A.Courville, and Y.Bengio.' 'Generative adversarial nets ". In Advances in neural information processing systems,2672-2680, 2014.

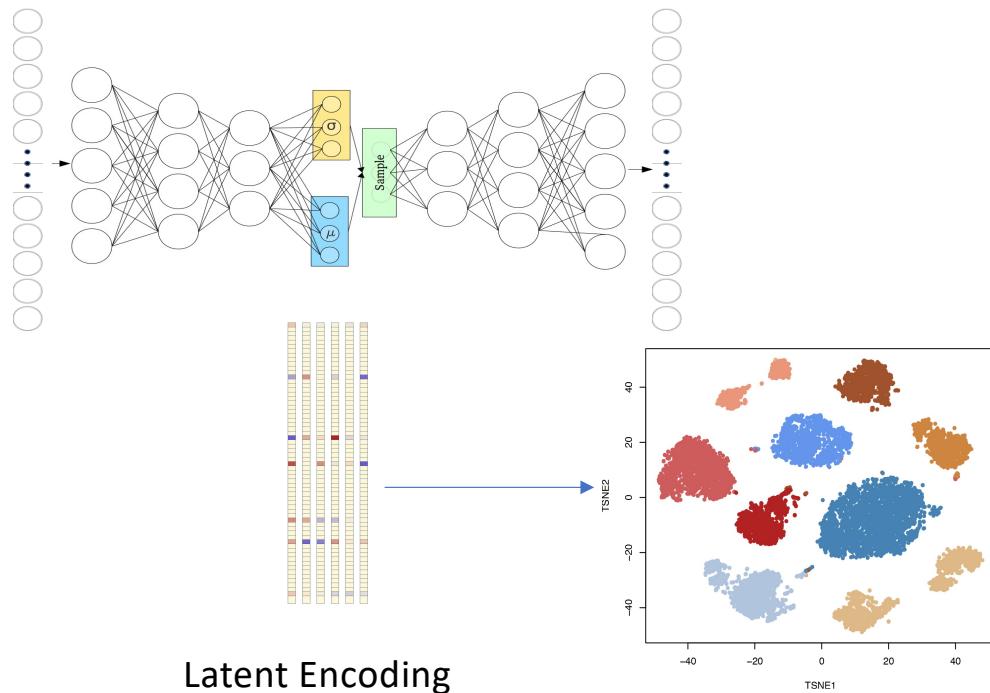


GANs have notoriously unstable training dynamics and suffer from what is known as “**mode collapse**”, which leads to some modes of the data being overrepresented and others missing.

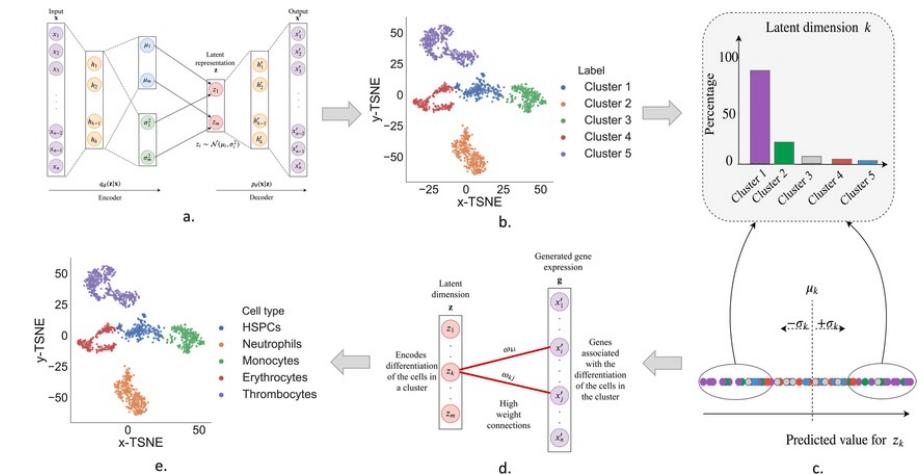
However, they are able to generate highly realistic “fake” samples

# Data visualization clustering and exploratory analysis

Gene Space



Gene Space

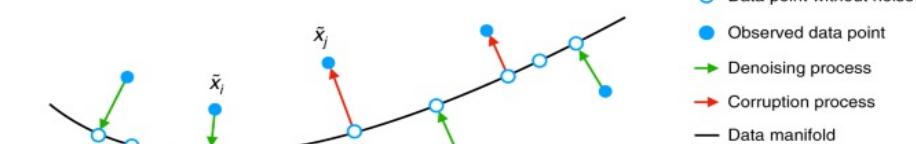


Unsupervised generative and graph representation learning for modelling cell differentiation  
nature research

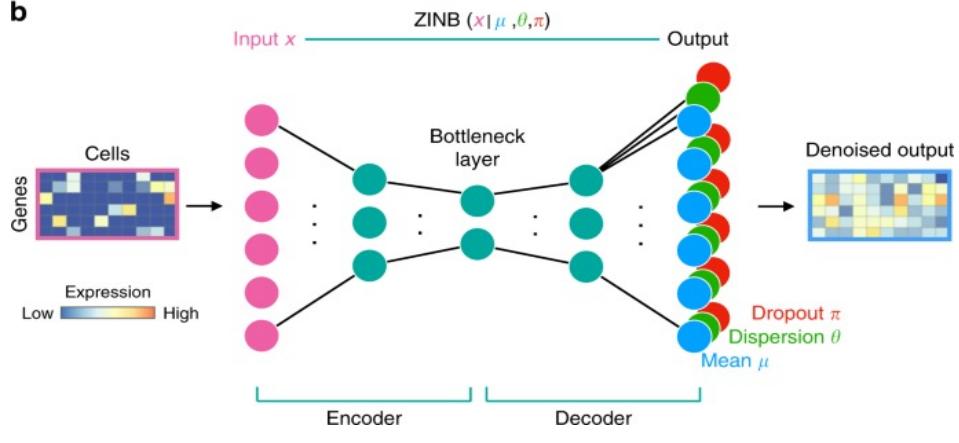
June 2020 · *Scientific Reports* 10(1):9790  
DOI:10.1038/s41598-020-66166-8

# Imputation and denoising

a



b



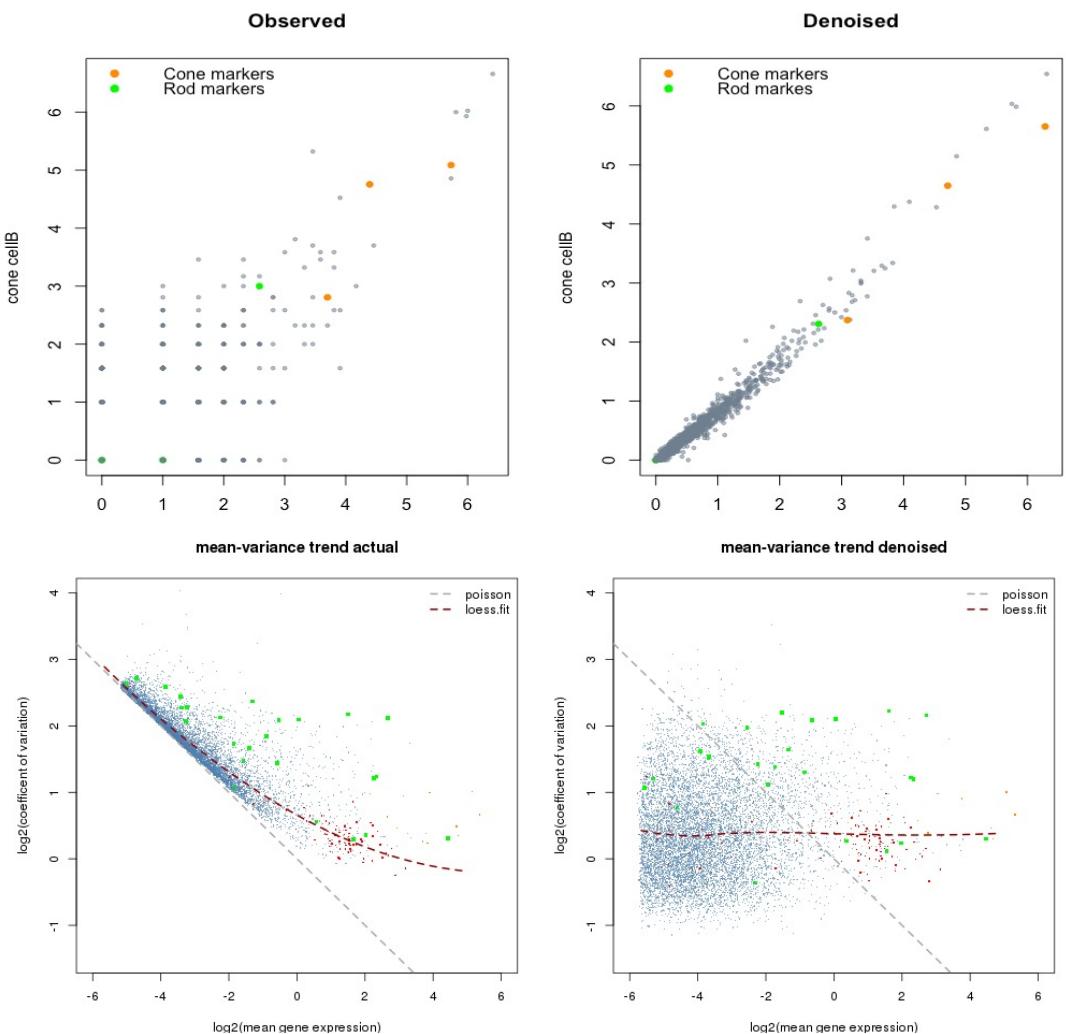
Article | Open Access | Published: 23 January 2019

## Single-cell RNA-seq denoising using a deep count autoencoder

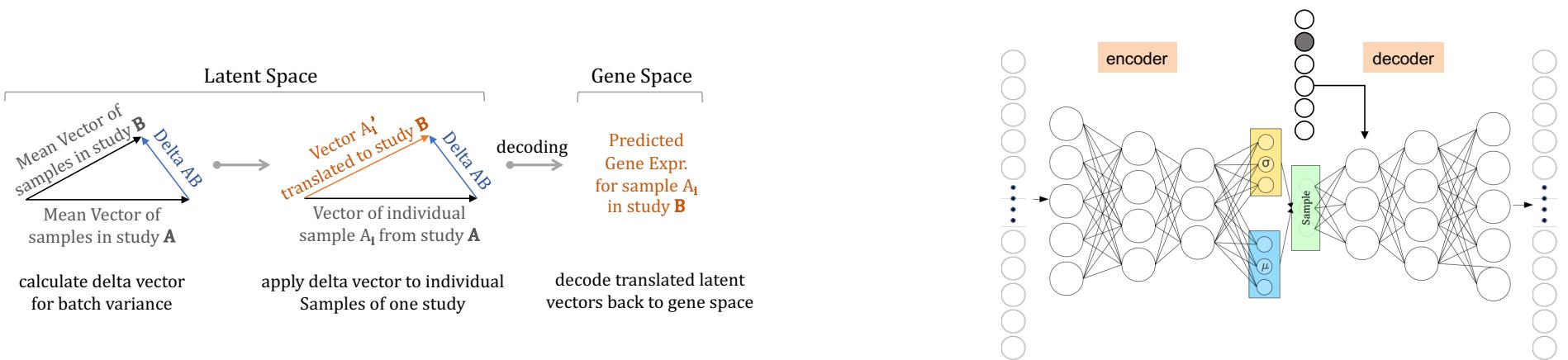
Gökçen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller & Fabian J. Theis

*Nature Communications* 10, Article number: 390 (2019) | Cite this article

- gimVI
- ScImpute
- Deep Count Autoencoder (DCA)
- DeepImpute



# Batch correction, data harmonization integration of heterogeneous scRNASeq data



## Exploring single-cell data with deep multitasking neural networks

Matthew Amodio, David van Dijk, Krishnan Srinivasan, William S. Chen, Hussein Mohsen, Kevin R. Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, Anita Desai, V. Ravi, Priti Kumar, Ruth Montgomery, Guy Wolf & Smita Krishnaswamy✉

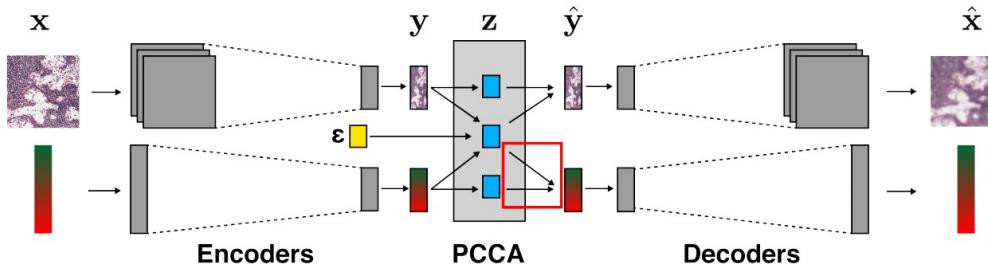
Nature Methods **16**, 1139–1145 (2019) | Cite this article

- SAUCIE
- scVI
- MAGAN
- CarDEC

# Multimodal data integration

## End-to-end training of deep probabilistic CCA on paired biomedical observations

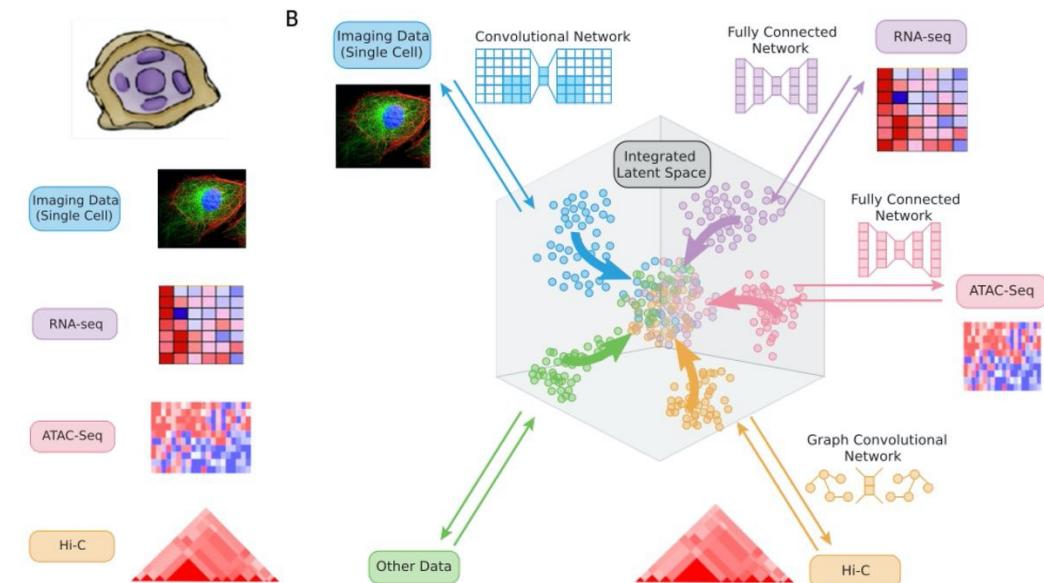
Gregory Gundersen \*  
gundersen@princeton.edu  
Bianca Dumitrascu †  
biancad@princeton.edu  
Jordan T. Ash \*  
jordanta@cs.princeton.edu  
Barbara E. Engelhardt  
bee@princeton.edu



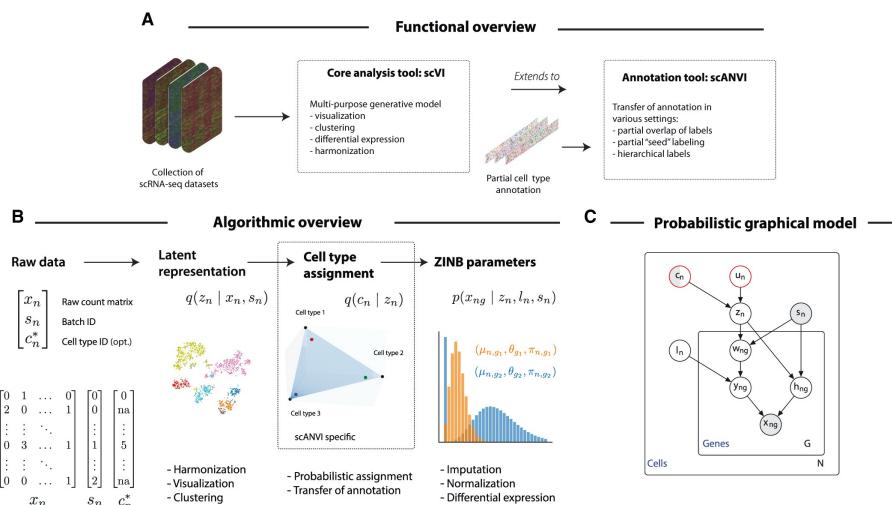
## Multi-domain translation between single-cell imaging and sequencing data using autoencoders

Karen Dai Yang, Anastasiya Belyaeva, Saradha Venkatachalam, Karthik Damodaran, Abigail Katcoff, Adityanarayanan Radhakrishnan, G. V. Shivashankar & Caroline Uhler [✉](#)

*Nature Communications* 12, Article number: 31 (2021) | Cite this article



# Automatic annotation of single cell data

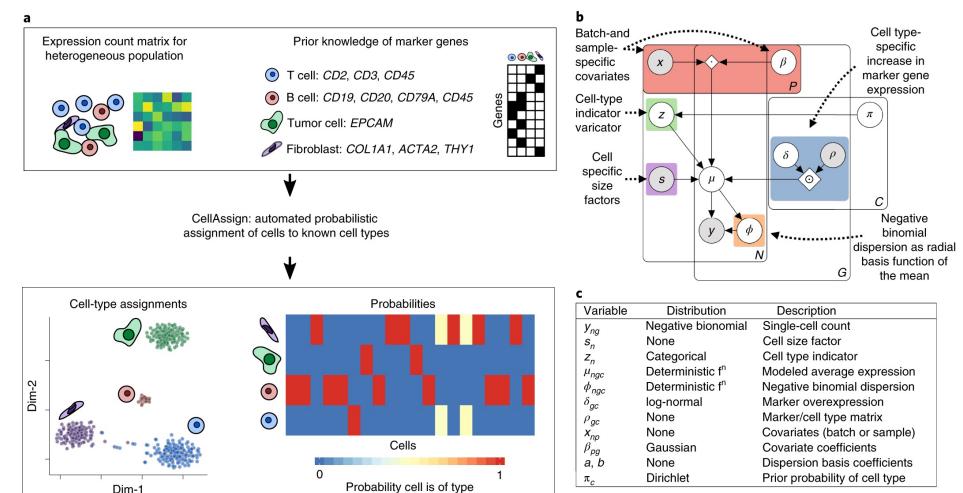


## Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models

Chenling Xu , Romain Lopez , Edouard Mehlman , Jeffrey Regier , Michael I Jordan , Nir Yosef

### Author Information

Mol Syst Biol (2021) 17: e9620 | <https://doi.org/10.15252/msb.20209620>

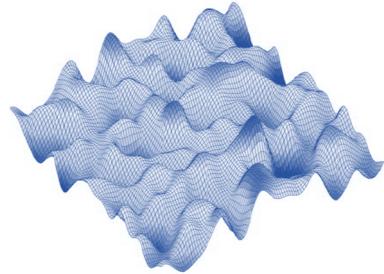


## Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling

Allen W. Zhang, Ciara O'Flanagan, Elizabeth A. Chavez, Jamie L. P. Lim, Nicholas Ceglia, Andrew McPherson, Matt Wiens, Pascale Walters, Tim Chan, Brittany Hewitson, Daniel Lai, Anja Mottok, Clementine Sarkozy, Lauren Chong, Tomohiro Aoki, Xuehai Wang, Andrew P Weng, Jessica N. McAlpine, Samuel Aparicio, Christian Steidl, Kieran R. Campbell & Sohrab P. Shah

Nature Methods 16, 1007–1015 (2019) | Cite this article

# DGN-based out-of-distribution inference on SC data



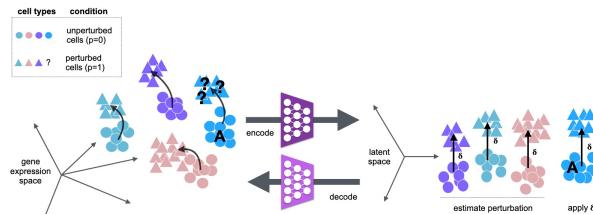
DGN based inference allows inspection of regions of the Tx landscape that have not been visited  
Some examples:

- Inferring transcriptomes upon biological perturbations (e.g in Silico KDs)
- Inferring effects of perturbations in different cell/tissue contexts (out-of-sample prediction)
- Inferring trajectories

## scGen predicts single-cell perturbation responses

Mohammad Lotfollahi, F. Alexander Wolf & Fabian J. Theis

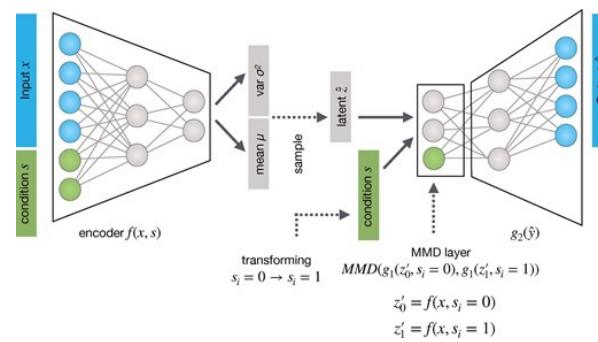
*Nature Methods* 16, 715–721(2019) | Cite this article



## Conditional out-of-distribution generation for unpaired data using transfer VAE FREE

Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J Theis, F Alexander Wolf

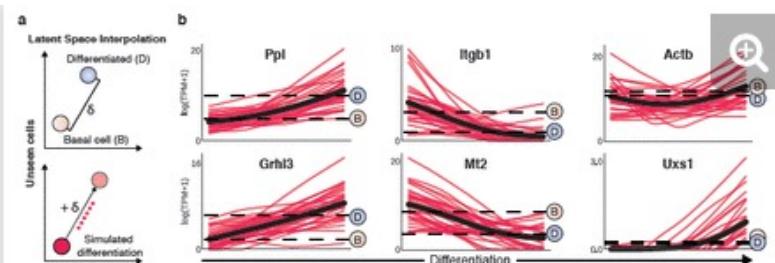
*Bioinformatics*, Volume 36, Issue Supplement\_2, December 2020, Pages i610–i617,  
<https://doi.org/10.1093/bioinformatics/btaa800>



## Generative adversarial networks uncover epidermal regulators and predict single cell perturbations

Arsham Ghahramani, Fiona M. Watt, Nicholas M. Luscombe

doi: <https://doi.org/10.1101/262501>

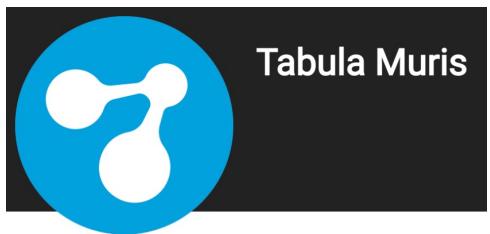


## Other applications

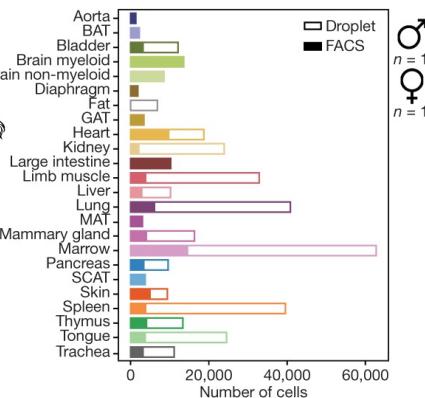
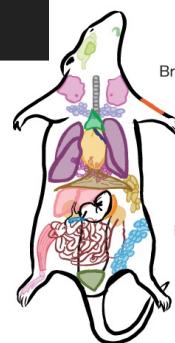
- Deconvolution of spatial transcriptomics data (Stereoscope, DestVI)
- Analysis of scATACseq data (peakVI)
- Doublet detection in scRNAseq data (Solo)
- Analysis of CITE-seq data (totalVI)
- Assessing gene specific levels of zero inflation (AutoZi)
- map query datasets on top of a reference (scArches)
- Gene regulatory networks inference (KPNNs)
- Deconvolution of bulk RNAseq data using scRNAseq atlases
- Rare cell detection
- In silico generation of datasets / data augmentation

# Group Project

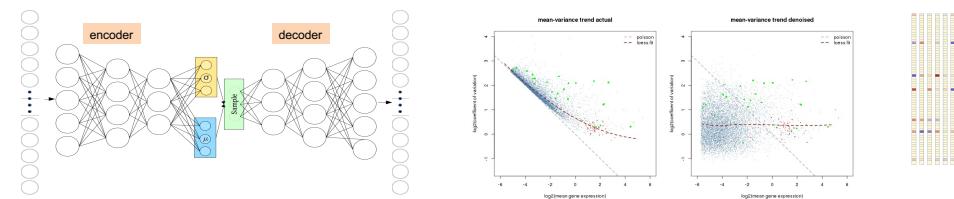
Model construction training, evaluation and use in exploratory analysis



Tabula Muris

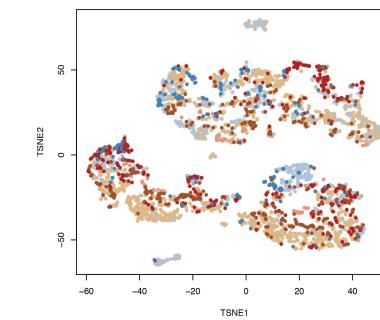
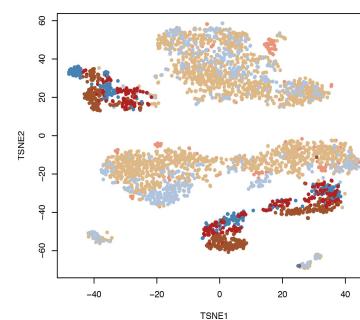


- Construct a single model for the provided dataset
- Training and model evaluation
- Use latent space for visualization. Explore latent variables.



Inference

- Assess the model's capacity for denoising (dropout imputation, outlier correction)
- Batch correction (due to use of the different technologies)
- Out-of-distribution prediction using latent arithmetic



## Perspectives

Despite the multitude of publications on DL in sc-omics the underlying principles are and used main architectures are relatively few.

Existing applications are not conceptual shifts but rather provide alternative implementations to problems that already have counterparts using different algorithmic approaches.

Geometric deep learning/structured learning: Graph convolutional networks

Allows for integration of existing biological knowledge in the network's inductive bias.

Sparser networks, more accurate representations

Methodology article | Open Access | Published: 08 July 2021

**Single-cell classification using graph convolutional networks**

Tianyu Wang, Jun Bai & Sheida Nabavi 

BMC Bioinformatics 22, Article number: 364 (2021) | [Cite this article](#)

1374 Accesses | 6 Altmetric | [Metrics](#)

Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data



August 2020 · *Genome Biology* 21(1)

DOI:10.1186/s13059-020-02100-5

Perturbation atlases combined with the representational capacity of DGNs hold the promise of more comprehensive mapping out of the regulatory manifold.

*Perturbation response prediction, Target and mechanism prediction, Prediction of combinatorial perturbation effects.*

*"After evaluating 6 classification methods across 14 datasets, we notably find that deep learning does not outperform classical machine-learning methods in the task... We, therefore, are still waiting for the "ImageNet moment" in single-cell genomics"*

Perspective

Machine learning for perturbational single-cell omics

Yuge Ji<sup>1,2</sup>, Mohammad Lotfollahi<sup>1,3</sup>, F. Alexander Wolf<sup>1,4</sup>, Fabian J. Theis<sup>1,2,4</sup>  

Deep learning does not outperform classical machine learning for cell-type annotation

Niklas D. Köhler<sup>\*1</sup>, Maren Büttner<sup>\*1</sup>, and Fabian J. Theis<sup>1,2</sup>