Deep Generative Networks and Variational inference in Single Cell Genomics

Panagiotis Papasaikas

FMI Computational Biology

Advanced SC course, Basel, 29-04-2022

Overview

- Motivation
- Introduction to Deep Learning* and High Level APIs for Deep Learning
- **Brief demontration of model building with Keras**
- Bias-variance decomposition, Representation Codes and DGN architectures
  - VAEs
  - GANs
- Applications in single cell-omics and existing tools
- **Building, evaluating and using a VAE for single-cell data with Keras/tf**
- **Demontration of scvi tools usage**

Why DGNs for transcriptomics ?

- **DGNs have the modelling capacity to capture and then be used for inference of very complicated real-world distributions without* any domain-expert knowledge of the underlying processes.**

- **Why it this important?**

- Often the underlying generative processes are poorly/incompletely understood: Explicitly specifying fully parametrized functional forms for these distributions is impossible.

- Classical ML models operate in underparametrized settings that limit their modelling capacity

- If we have in our hands a good* model of the generative process, several tasks become possible in a unifying setting:

New instance generation, feature summarization and abstraction, classification, segmentation, denoising/imputation/completion, out-of-sample prediction, integration and translation across "styles", batches, modalities.

Importantly in the modern setting DL models are specified and trained end-to-end.
Training instances → Model → Evaluation against a cost function

Is the use of DGNs in the context of SC analysis even necessary / over the top?
(DL for the sake of DL)

→ It depends on the context:

When operating in the strict context of one/few datasets, one/few experiments and specific hypotheses then ”classical” approaches are just fine.
The advantage of DGNs comes when we are faced with problems that force us to move beyond the data at hand. (working across modalities, inference across cellular or molecular contexts, prior knowledge integration, new instance generation)

DL models are black boxes that curtail interpretability

→ Not true. Several methods exist to look under-the-hood.
→ Interpretable features can be explicitly built in the model’s inductive bias .
→ Sometimes interpretability is not of essence.

DGNs are over-parametrized, non generalizable overfitting machines.

→ This has been comprehensively refuted across multiple domains.
Modern ML theory has moved beyond the classical bias-variance tradeoff.

Why should I care?

It's true that in simple SC analysis settings DL does not offer overwhelming advantages.

It is also true that the field is in a state of extreme flux which can be demoralizing:
A lot of "experimentation", tough to keep-up, identify useful methods, benchmarking and efforts to integrate in existing workflows are not where they should be.

But...

Experimental designs are becoming more intricate.

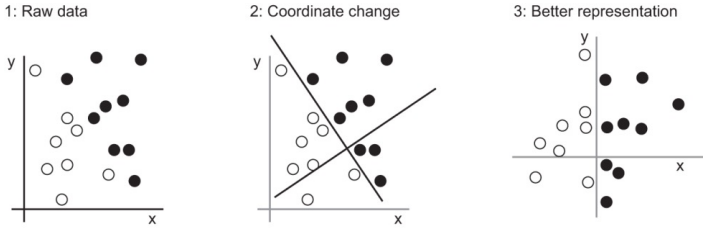
Integration of very diverse datasets/atlases, modalities is more and more common.

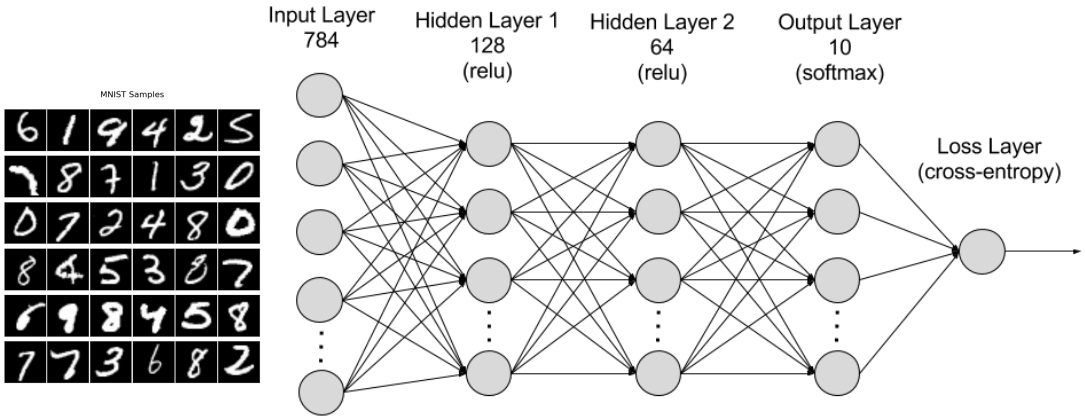
Interpretable models i.e putting SC experiments in the context of prior biological knowledge (gene pathways, disease, biological programs) are important.

Inference across cell types, states, is essential especially in R&D.

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

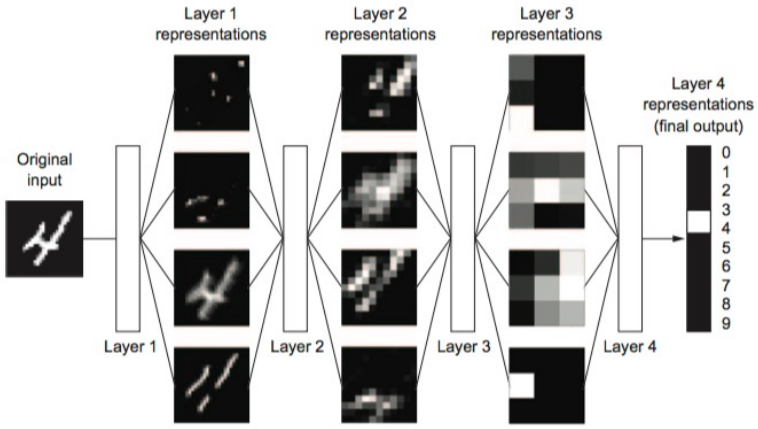


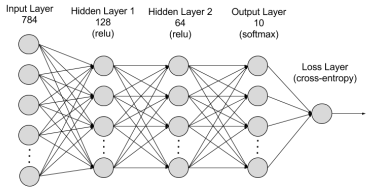
Figure 1.6 Deep representations learned by a digit-classification model

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



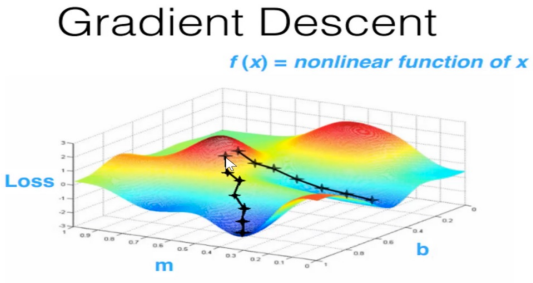
The l**oss function** measures the success of the model for the task at hand.

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

optimizer

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 (SGD)** variants



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

What spurred the revolution?

Mainly advances on three fronts:

- **Massively parallel computation hardware** (GPUs, TPUs)

- **Improved algorithms**

robust backprop, optimizers, regularization techiniques

- **Large, high-quality (often labeled) datasets**

web usage, advances in tech/instrumentation in hard sciences

↓

Improved architectures

User-friendly platforms

Successes of Deep Learning

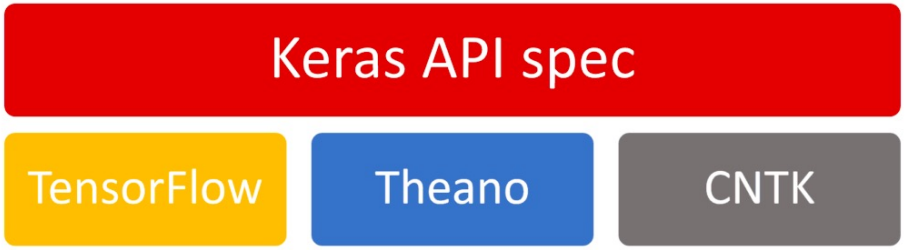
- Refined web-searching
- Spam/Fraud detection
- Near-human image classification (**MSRA, ImageNET**)
- Near-human machine translation (**DeepL**)
- Superhuman chess/GO playing (**AlphaZero**, **LC0**)
- Autonomous driving
- Natural language processing (e.g **IBM debater**, **GPT-x, PALM, Chinchilla**)

- Protein Folding (AlphaFold)
- Medical Image Processing
- Drug design
- Diagnostics

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

- TF is an open source general purpose numerical computing library (not only DL, e.g general optimization libraries).
- Originally developed by engineers in the Google Brain Team for conducting ML research
- Hardware independent (CPUs, GPUs, TPUs)
- Supports large datasets/distributed execution

Keras as a high level API supports multiple DL backends:



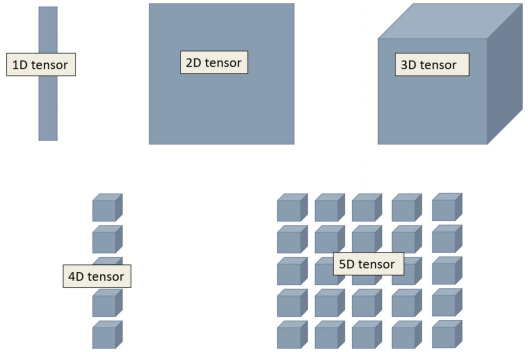
Multiple Deep Learning frameworks:

The model building blocks in Tensorflow/Keras/Pytorch

- **Tensors** are multidimensional arrays.

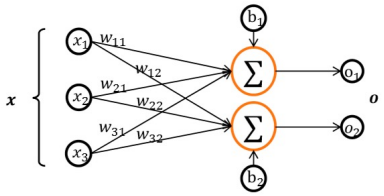
Data	Tensor dimension	R object
Cell label	1D (samples)	vector
Gene Count Matrix	2D (samples, genes)	matrix
Longitudinal data	3D (samples, genes, timestamp)	3d array
Microscopy Images	4D (samples, height, width, channels)	4d array
Video	5D (sample, height, width, channel, frame)	5d array

*Notice the orientation convention is opposite to what 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…

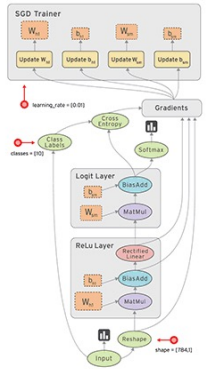


- Layers and Tensors are combined to contruct computation **graphs** (DAGs).

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

Keras

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

```r
#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')
```

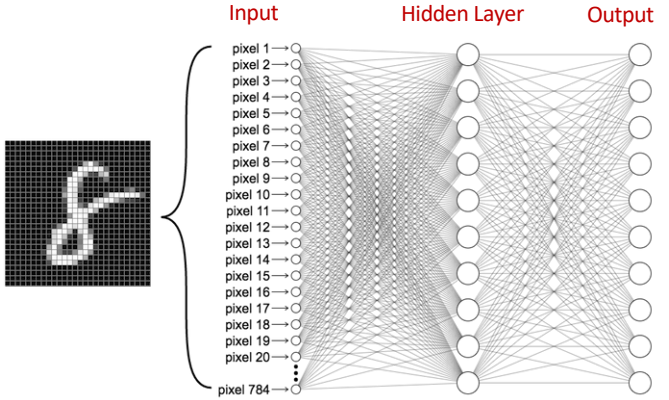
```python
#defining model with one input layer[784 neurons], 1 hidden layer[784 neurons] with dropout rate 0.4 and 1 output l
model=Sequential()

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'))
```

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

relu activation

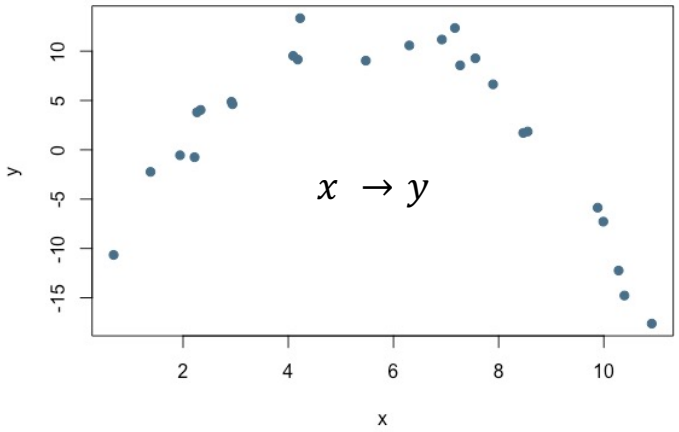
$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

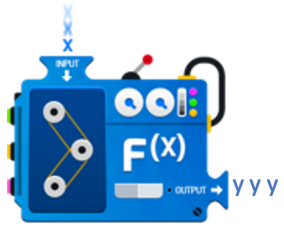
sotfmax activation



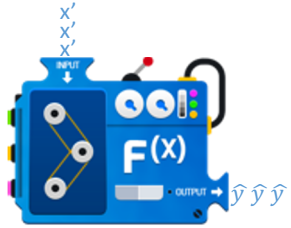
MLP model for digit classification

Bias-Variance decomposition

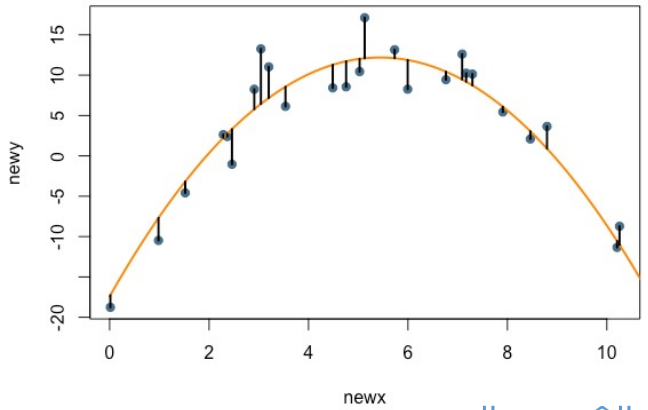




Train



Test



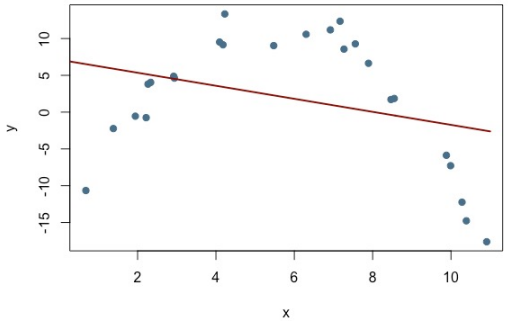
$$Prediction\ Error \simeq \left\| Y' - \hat{Y} \right\|$$

$$Prediction\ Error = BiasError + VarianceError + Irred.Error\ (BayesError)$$

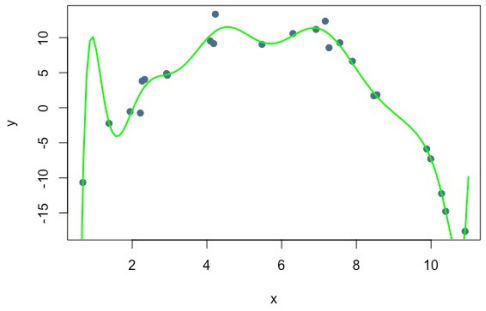
Modelling assumptions (too rigid)

High sensitivity (not rigid enough)

Noise



High bias



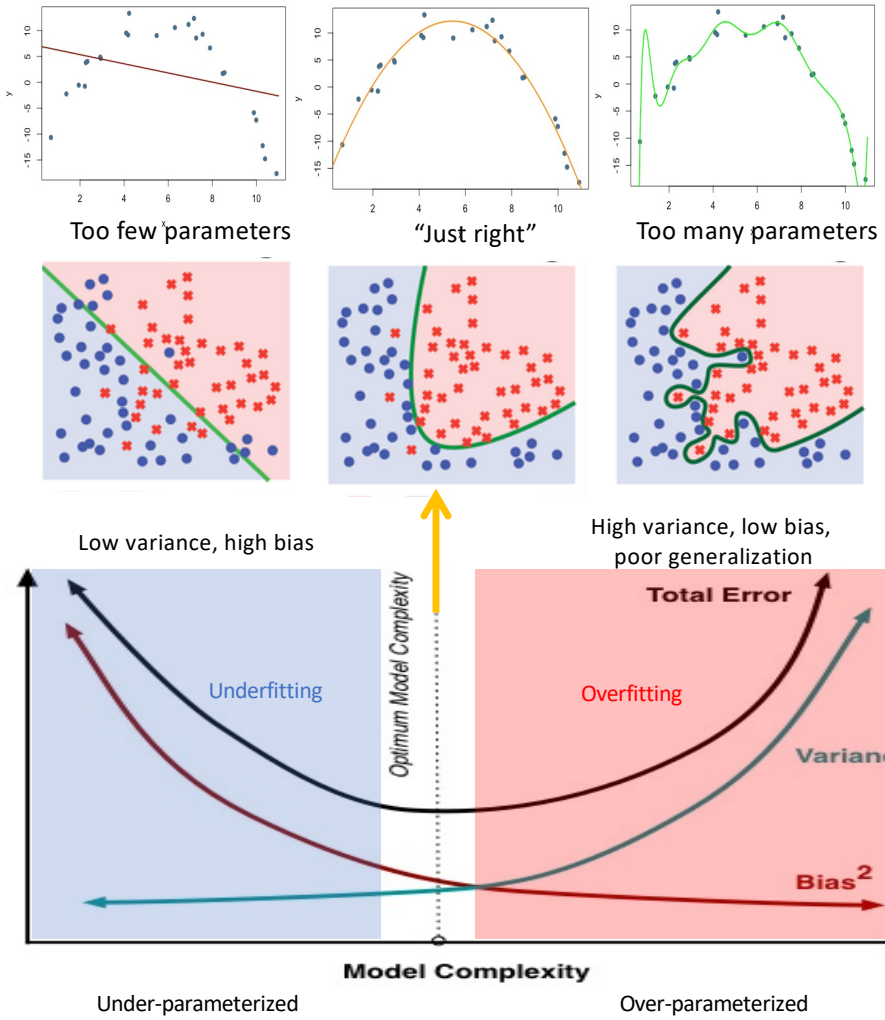
High variance

High Training Error → Bias (Underfitting)

Low Training Error
High Test Error → Variance (Overfitting / Loss of Generalization)

Only Irred. Error → Bayes limit

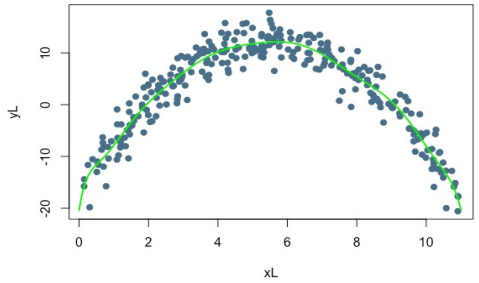
Bias-Variance tradeoff



"As the model capacity increases, so does its variance, its potential to overfit and its generalization error"

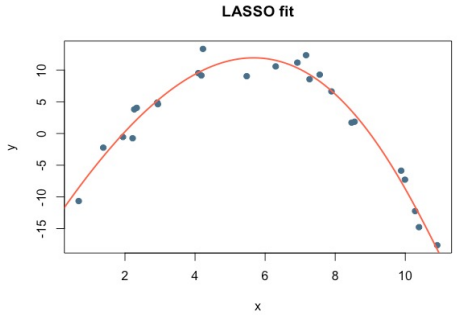
So how do we increase the number of model parameters (p) while preventing overfitting?

- Increase training set ($n >> p$)



- Use **regularization**

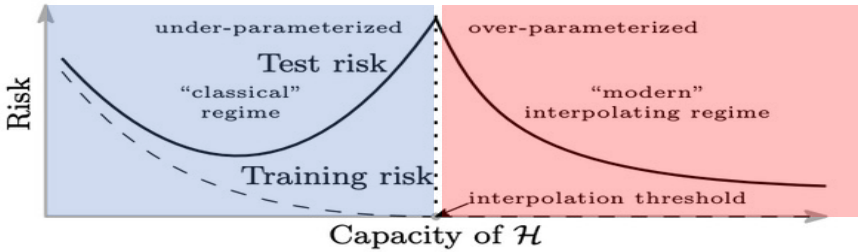
(penalize for the number and/or for the magnitude of the model parameters)



Expands the selection of model class while keeping the **effective model complexity** low

What are the mechanics that drive performance improvement in the overparameterized regime?

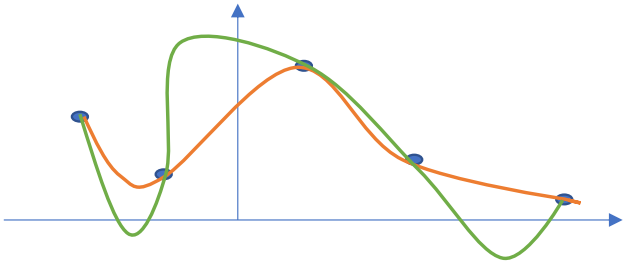
- Modern deep learning models are hugely overparameterized ($p >> n$)
- Actually in several instances they appear to be completely overfitted (even in the presence of noisy / mislabelled data)
- However, their generalization performance is extremely good



The "double descent" curve

- Variance appears to decrease with model complexity beyond the IP
- Overfitting is not necessarily harmful
- Since the IP threshold is at $p=n$ does that mean that in some cases more data can hurt us?

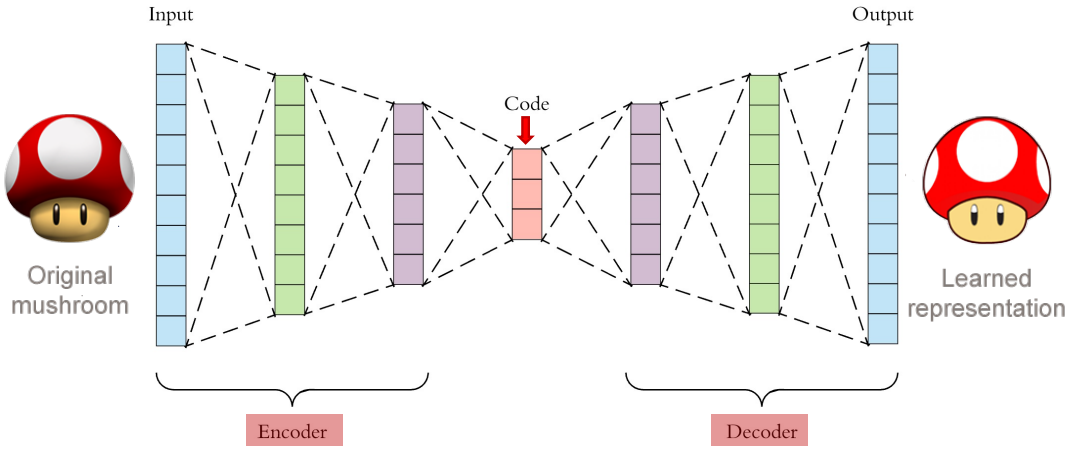
- No violation of the spirit of bias-variance tradeoff. More complex models DO have a higher variance upper-bound but that bound is not realized.
- It's the dynamics of the exploration of the optimization landscape by **SGD** that lead to "smoother" solutions with low weight norms.
- These solutions are more robust to input fluctuations and thus generalize well

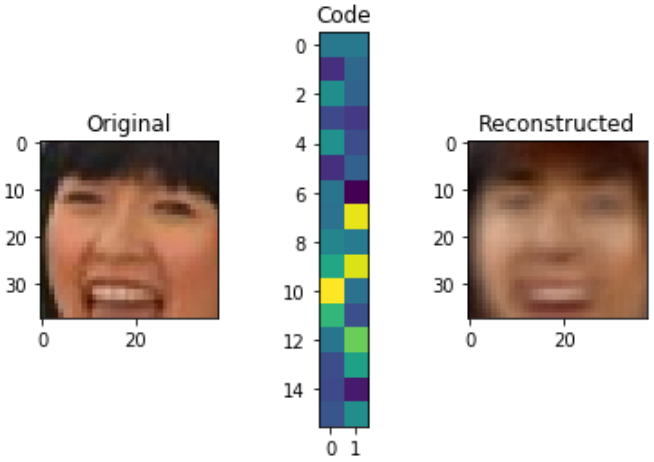


- Even in the cases of complete overfitting, the solutions are (benign)
- Overparametrization is essential for "**benign**" overfitting.
- This effect of the specific optimization procedure is referred to as "**implicit regularization**".
- In a way analogous to classical regularization it expands the selection of model class while keeping the **effective model complexity (intrinsic dimension)** low.

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:



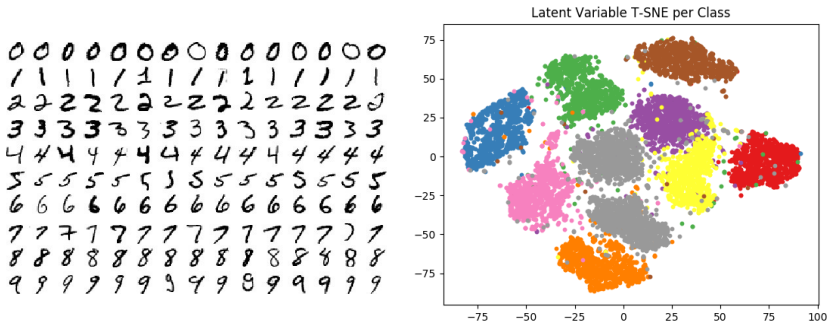


Multiple AE flavors

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

Applications of AEs

1. Dimensionality reduction & visualization

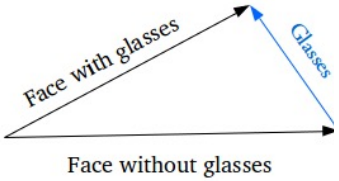


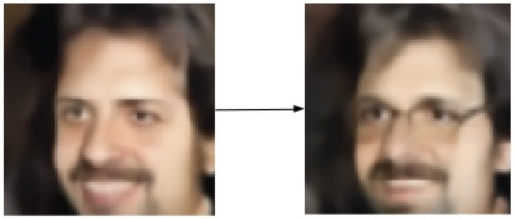
2. Denoising & completion (imputation)

Digit Denoising

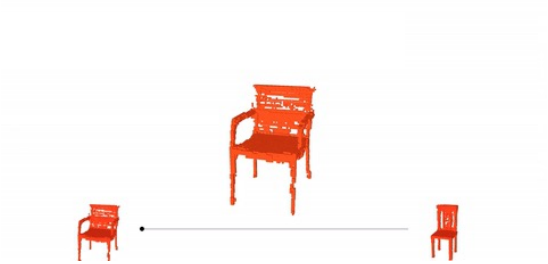
Face completion

3. Feature manipulation , interpolation and exploration





Subject Subject + Glasses



Multiple AE flavors

Deep/Stacked, Sparse, Variational, Denoising, Adversarial, Disentangling…

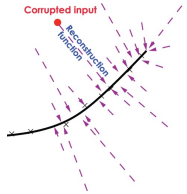
Why AEs for SC transcriptomics?

Tx data: High dimensional Noisy/corrupt ➞ ➞ Visualization Denoising

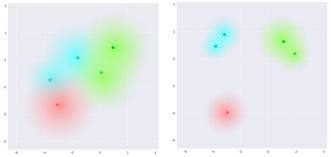
Latent representations and “good” representation codes

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

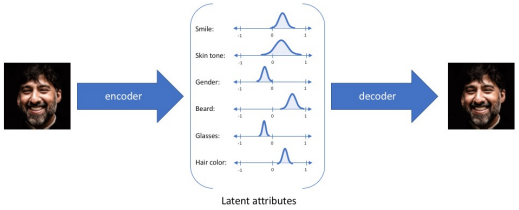
- Robust to “meaningless” input corruptions



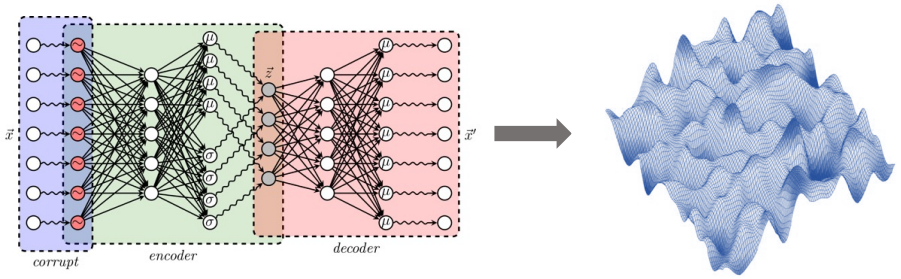
- Generalizable ⇒ can transfer to multiple settings /related problems
- Smooth / Coherent: similar inputs ⟼ similar codes.



- Explanatory

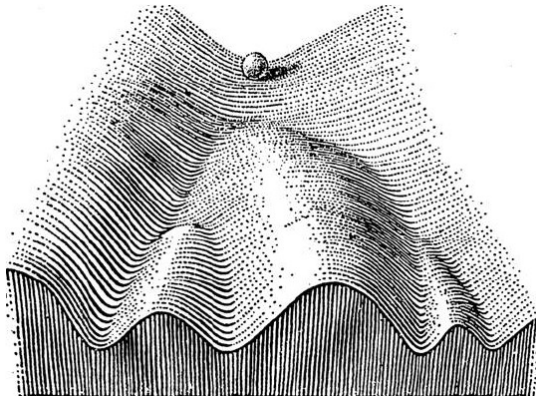


The latent representation is an estimation of the unerdlying **manifold** that gives rise to the data

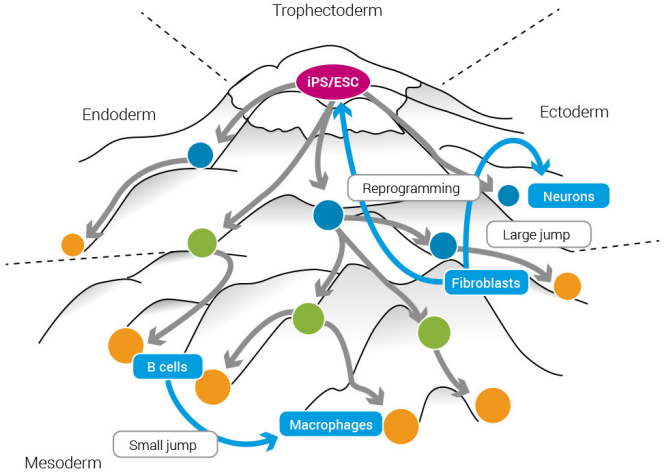


- Succinct, generative representations of complex Tx manifolds.
- Each location in this manifold represents a different realizable cell-state

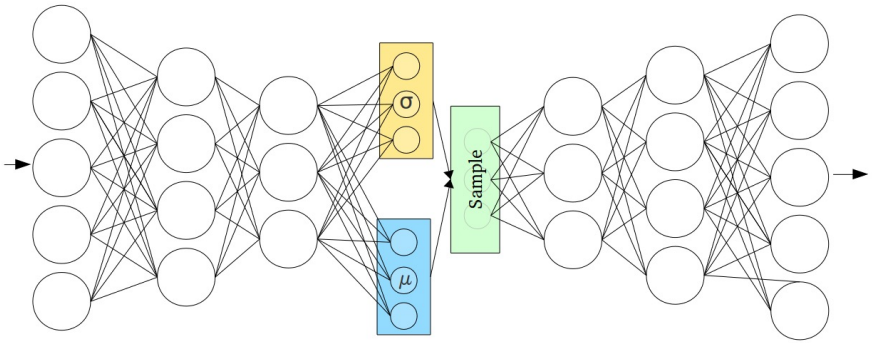
A useful analogy:



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.

$$\mathcal{L}_\beta = \frac{1}{N}\sum_{n=1}^{N} \underbrace{\left(\mathbb{E}_q[\log p(x_n|z)]\right)}_{\text{Reconstruction}} - \underbrace{\beta\, \mathrm{D}_{\mathrm{KL}}\left(q(z|x_n)||p(z)\right)}_{\text{Distance to latent prior}}$$

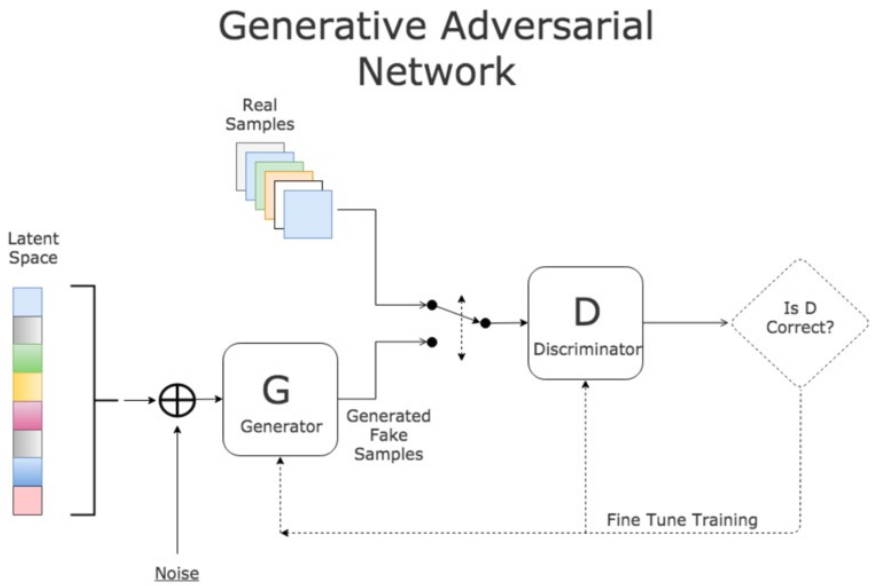
The latent prior is a multivariate normal with a unit covariance matrix

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

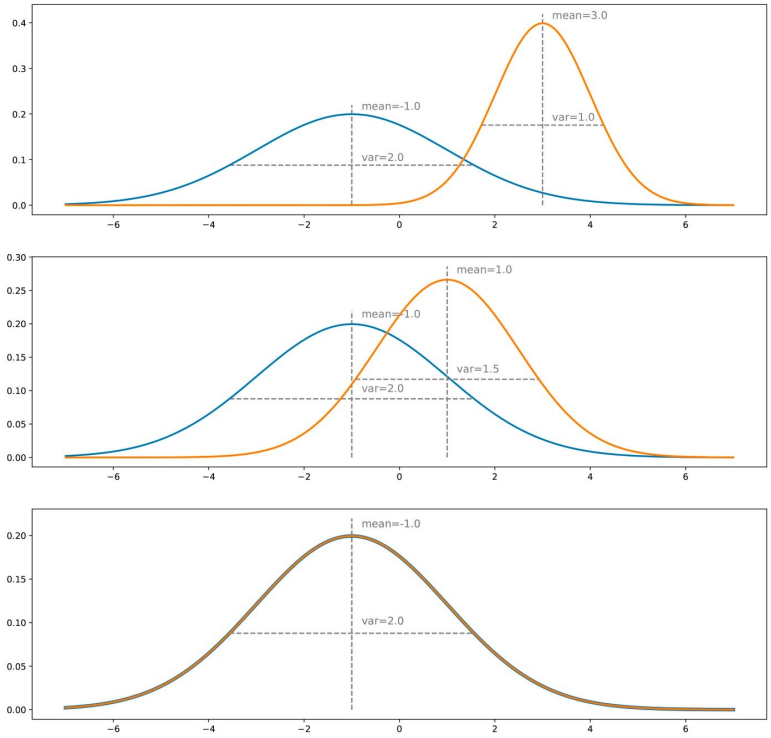
Most models used in SC-omics are based on VAE with modifications in:

- **The model inductive bias**
- **The optimization function**

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.

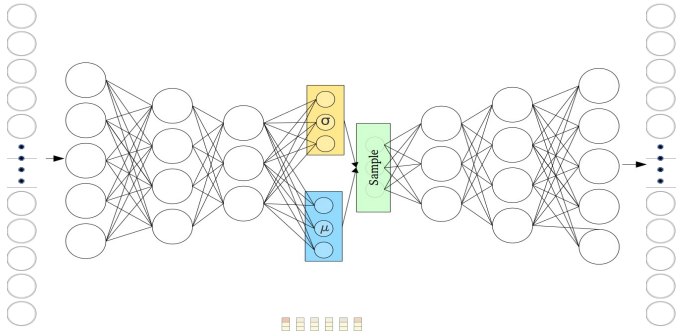
However, they are able to generate highly realistic "fake" samples

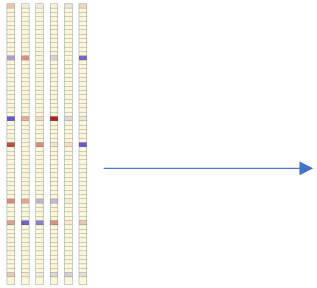
Questions so far?

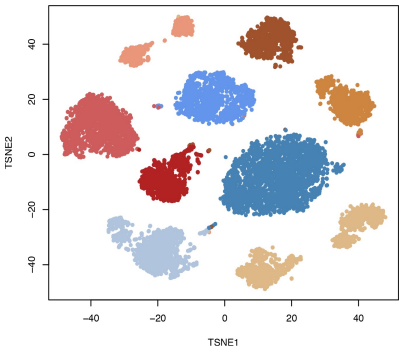
Data visualization clustering and exploratory analysis

Gene Space

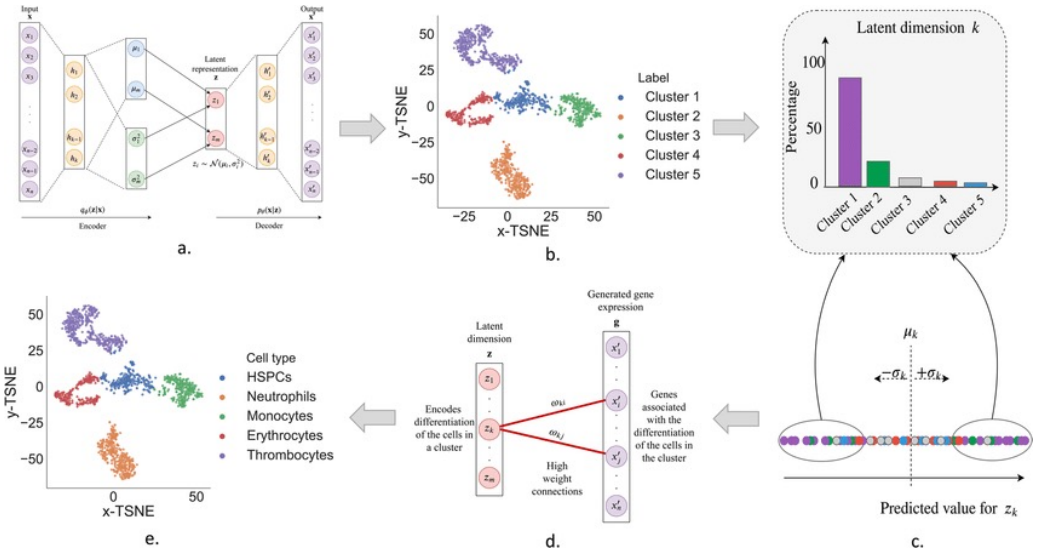
Gene Space







Latent Encoding



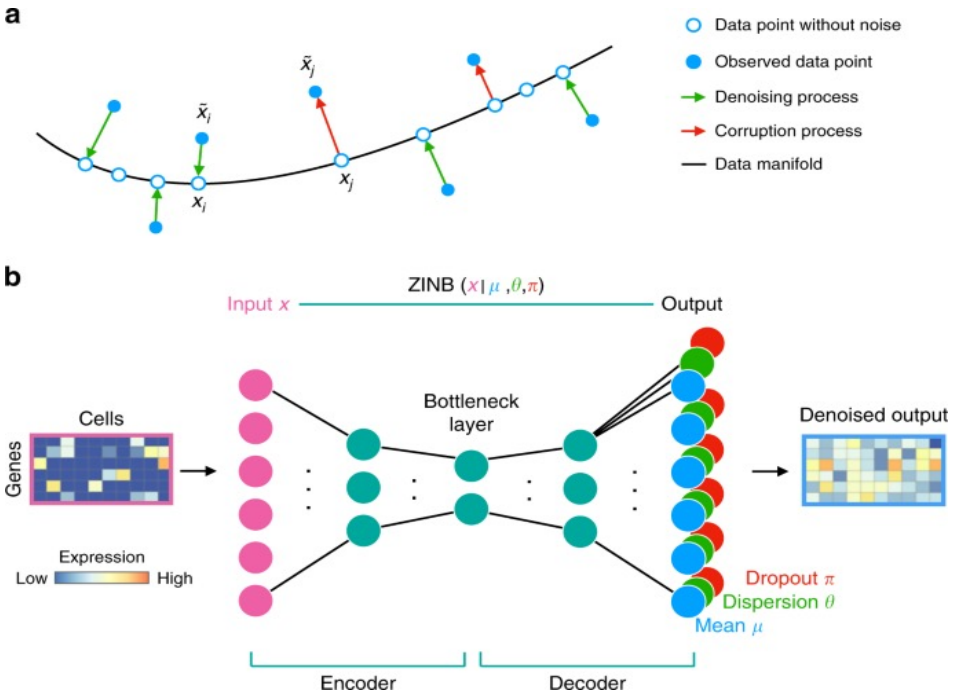
Unsupervised generative and graph representation learning for modelling cell differentiation

nature research

June 2020 · Scientific Reports 10(1):9790

DOI:10.1038/s41598-020-66166-8

Imputation and denoising

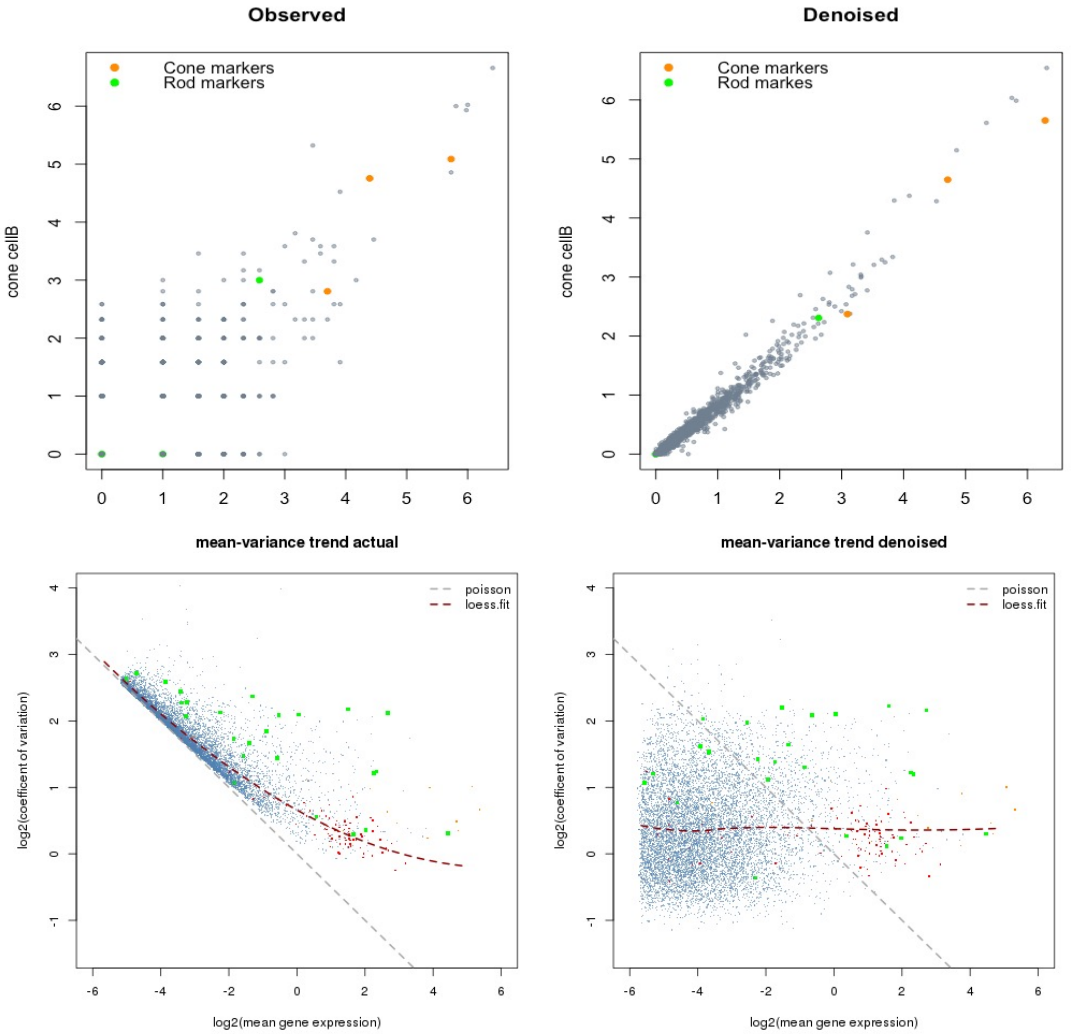


Article | Open Access | Published: 23 January 2019

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

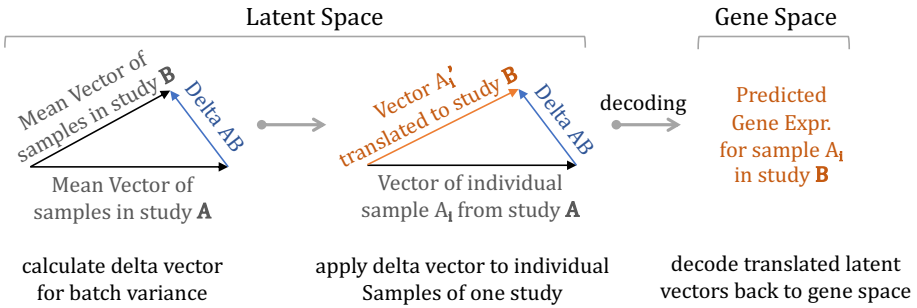
Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller & Fabian J. Theis

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

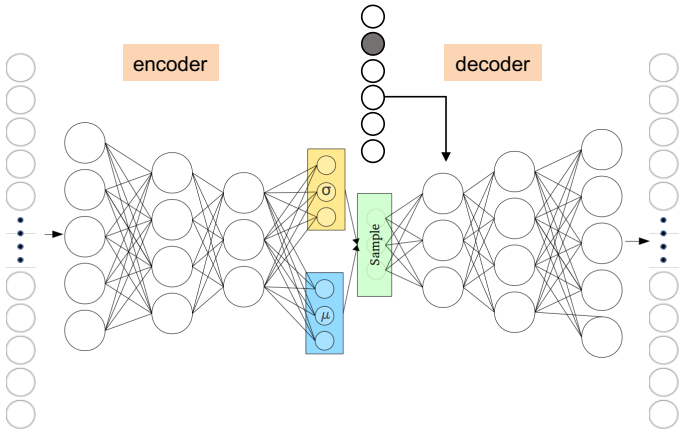


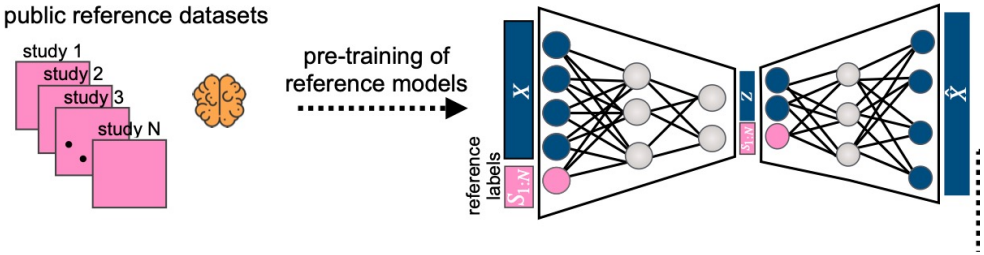
- gimVI
- DeepImpute
- ScImpute
- Deep Count Autoencoder (DCA)

Batch correction, data harmonization integration of heterogeneous scRNAseq data



Laten arithmetic based

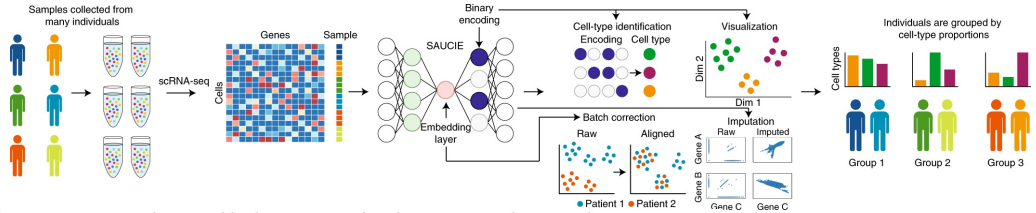




Deep generative modeling for single-cell transcriptomics

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan & Nir Yosef

Conditional Variational Autoencoders



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

Style transfer

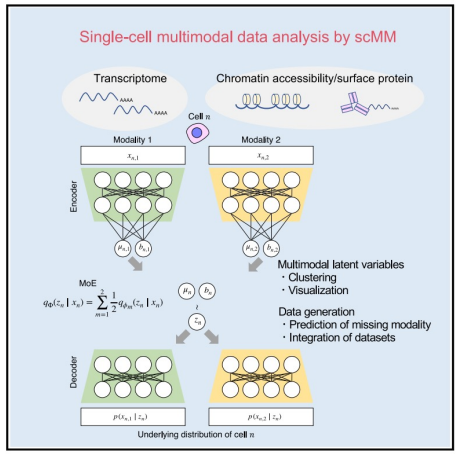
- SAUCIE
- scVI/scARCHES
- MAGAN
- CarDEC

Multimodal data integration

Cell Reports
Methods

Article

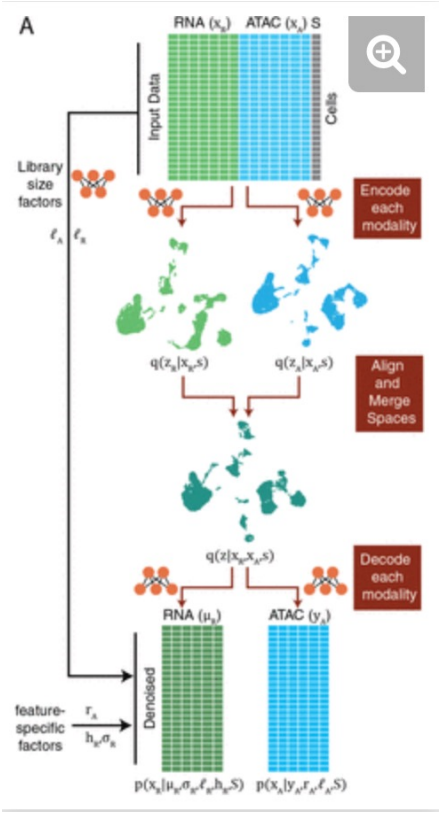
A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data



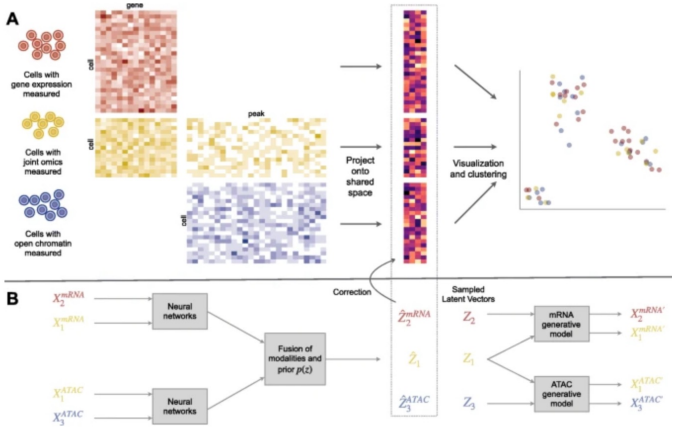
MultiVI: deep generative model for the integration of multi-modal data

Tal Ashuach, Mariano I. Gabitto, Michael I. Jordan, Nir Yosef

doi: https://doi.org/10.1101/2021.08.20.457057



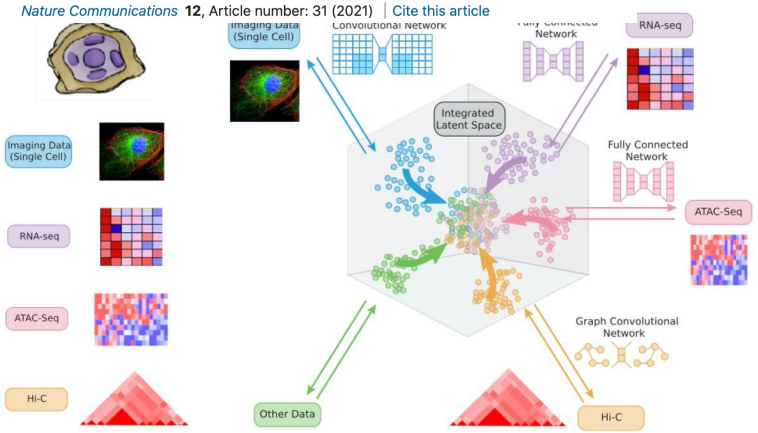
Cobolt: integrative analysis of multimodal single-cell sequencing data



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

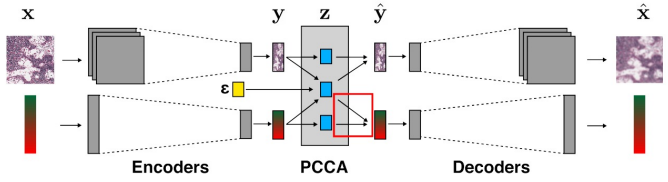
Karren Dai Yang, Anastasiya Belyaeva, Saradha Venkatachalapathy, Karthik Damodaran, Abigail Katcoff, Adityanarayanan Radhakrishnan, G. V. Shivashankar & Caroline Uhler

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



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

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



NeurIPS 2021

Multimodal single cell data integration challenge: results and lessons learned

https://www.biorxiv.org/content/10.1101/2022.04.11.487796v1.full

Bechmarking datasets:
CITE-seq (GEX + ADT): 90K cells
10x Multiome (ATACseq + GEX): 70k cells

- **Task 1: Cross-modal prediction (predict one modality from another)**

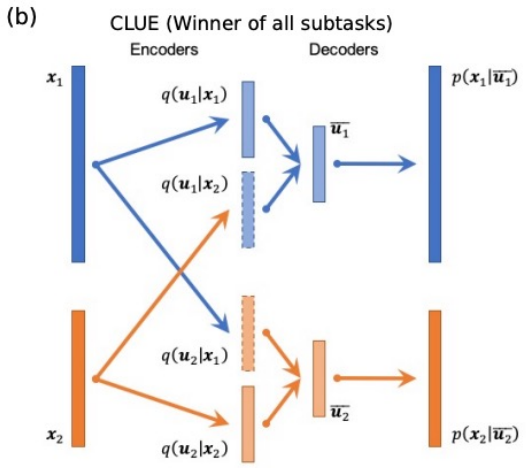
Most high scoring taks were DGN-based. Relatively poor performance especially in the GEX2ATAC, ATAC2GEX regime.

- **Task 2: Match cells between modalities (cell-alignment)**

One clear winner across all subtasks (Cross-linked universal embedding, CLUE):
VAE with a shared embedding resulting from concatenation of modality-specific subspaces.

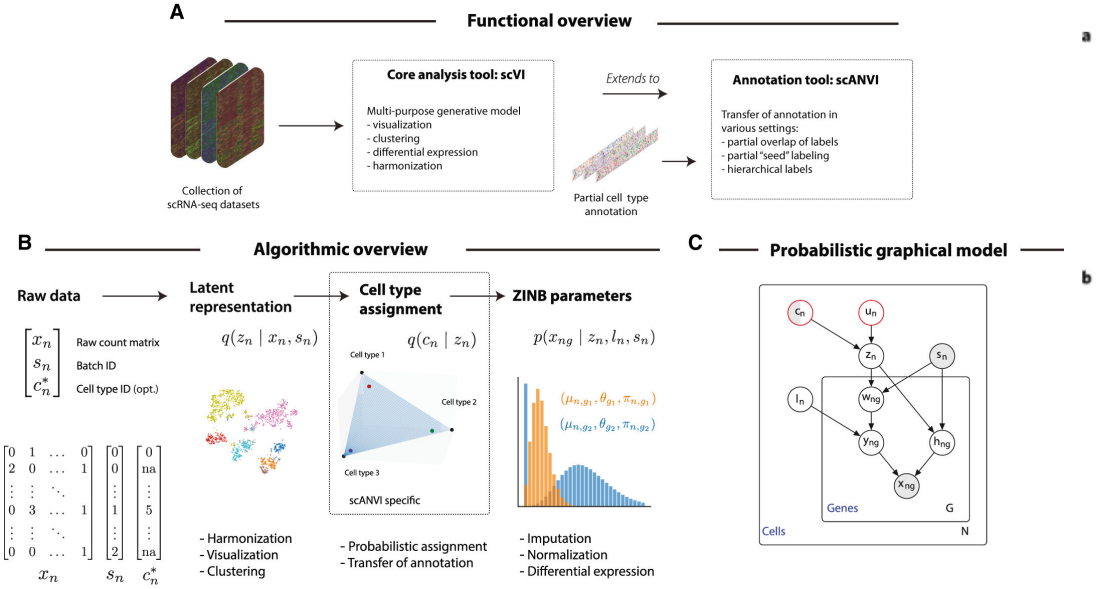
- **Task 3: Learn joint representations (manifold alignment)**

Joint embedding with a regularized autoencoder (JAE)



Automatic annotation of single cell data

scANVI



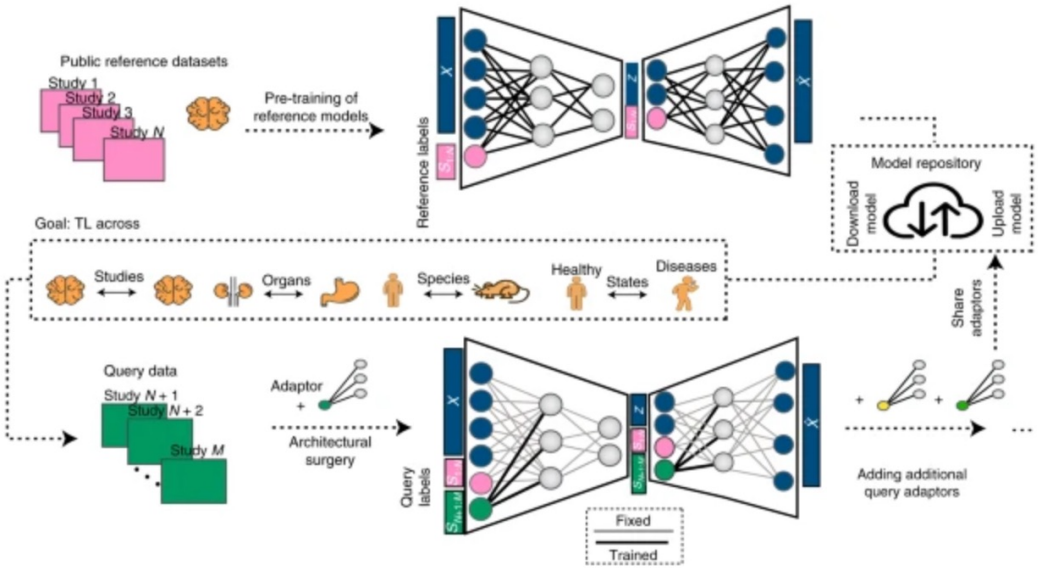
Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models

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

scArches



Mapping single-cell data to reference atlases by transfer learning

Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin & Fabian J. Theis

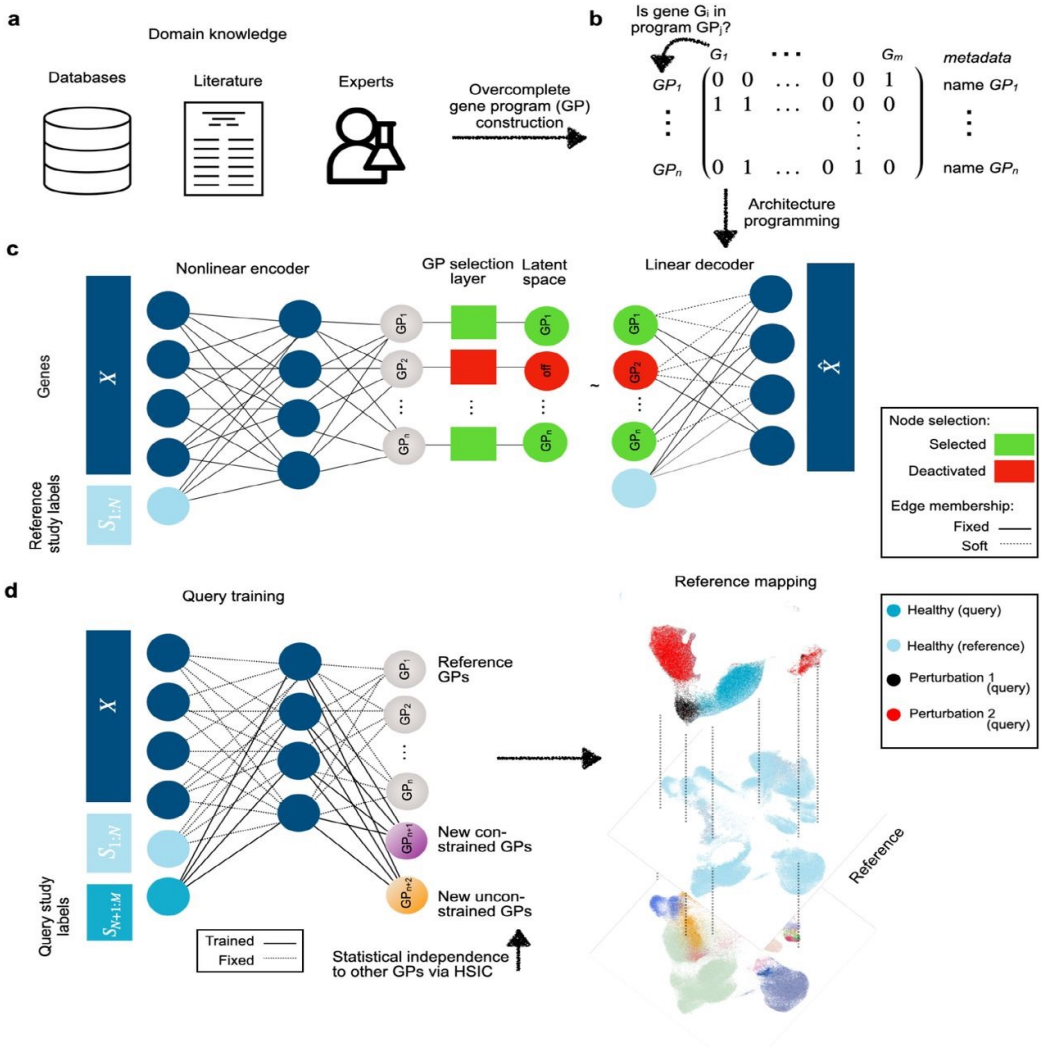
Interpretable models
(contextualize relative to prior biological knowledge e.g gene pathways, disease, biological programs)

Biologically informed deep learning to infer gene program activity in single cells

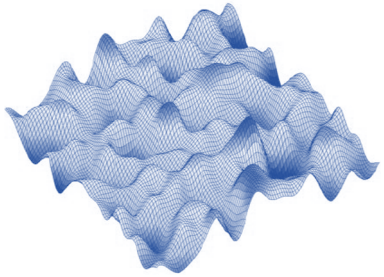
Mohammad Lotfollahi, Sergei Rybakov, Karin Hrovatin, Soroor Hediyeh-zadeh, Carlos Talavera-López, Alexander V Misharin, Fabian J. Theis

doi: https://doi.org/10.1101/2022.02.05.479217

expiMap



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

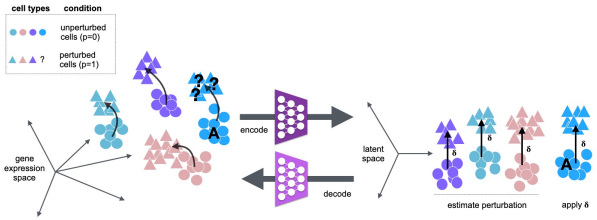
Some examples:

- 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

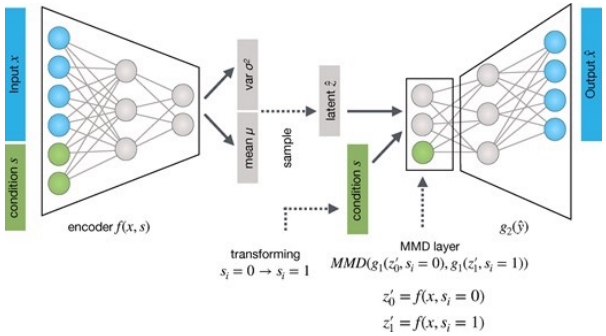
Nature Methods **16**, 715–721(2019) | Cite this article



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

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

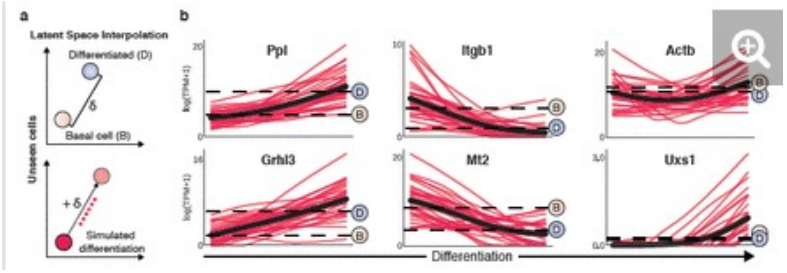
Bioinformatics, Volume 36, Issue Supplement_2, December 2020, Pages i610–i617, https://doi.org/10.1093/bioinformatics/btaa800



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

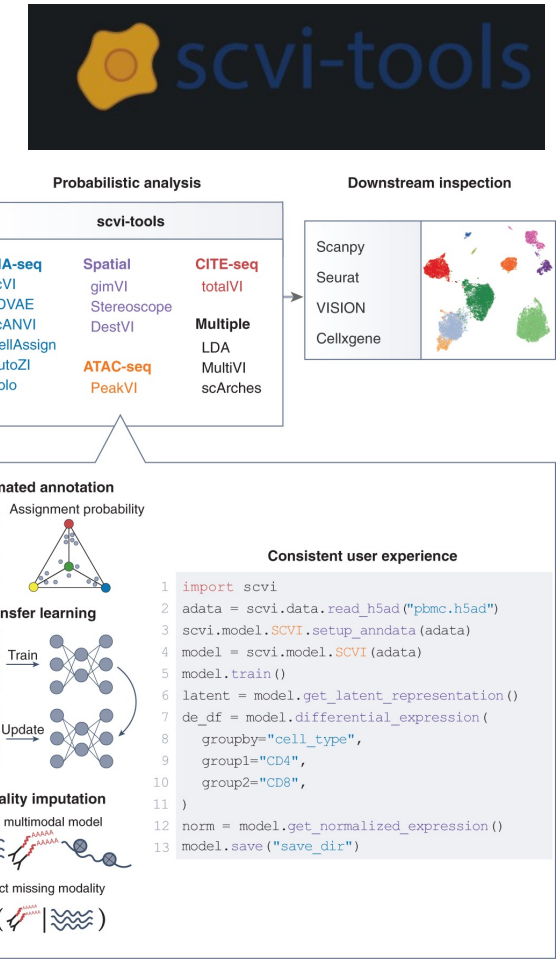
- Deconvolution of spatial transcriptomics data (Stereoscope, DestVI)
- Analysis of scATACseq data (peakVI)
- Doublet detection in scRNAseq data (Solo)
- Analysis of CITE-seq data (totalVI)
- Assessing gene specific levels of zero inflation (AutoZi)
- Gene regulatory networks inference (KPNNs)
- Deconvolution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

scvi tools

A Python library for probabilistic analysis of single-cell omics data

Adam Gayoso, Romain Lopez, ... Nir Yosef + Show authors

Nature Biotechnology **40**, 163–166 (2022) | Cite this article



https://scvi-tools.org/

Summary/Perspectives

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

The field is very dynamic and fast growing. Can be tough to keep-up / identify important contributions

Many applications are not conceptual shifts but rather provide alternative implementations to problems that already heave counterparts using different algorithimic approaches.

DGNs excel in generalization/abstraction/contextualization/new instance generation/inference/modality integration

Some upcoming trends:

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

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

BMC

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 compreh mapping out of the regulatory manifold.
Perturbation response prediction, Target and mechanism prediction, Prediction of combinatorial perturbation e

Perspective

Machine learning for perturbational single-cell omics

Yuge Ji [1, 2], Mohammad Lotfollahi [1, 3], F. Alexander Wolf [1, 4], Fabian J. Theis [1, 2, 4]