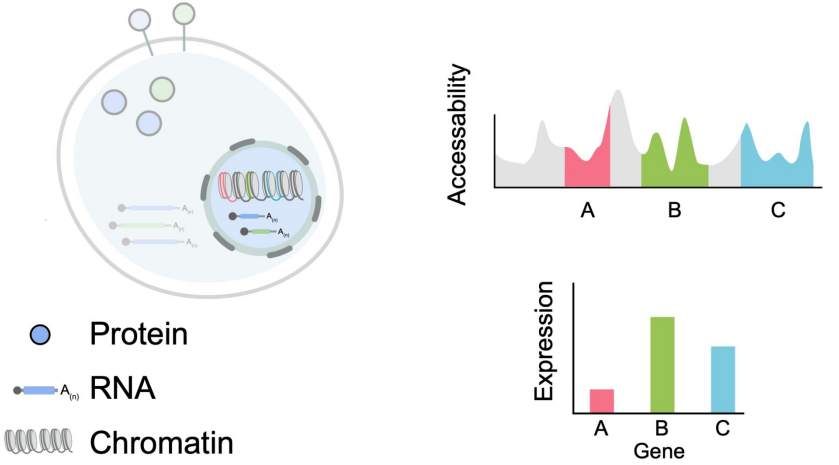
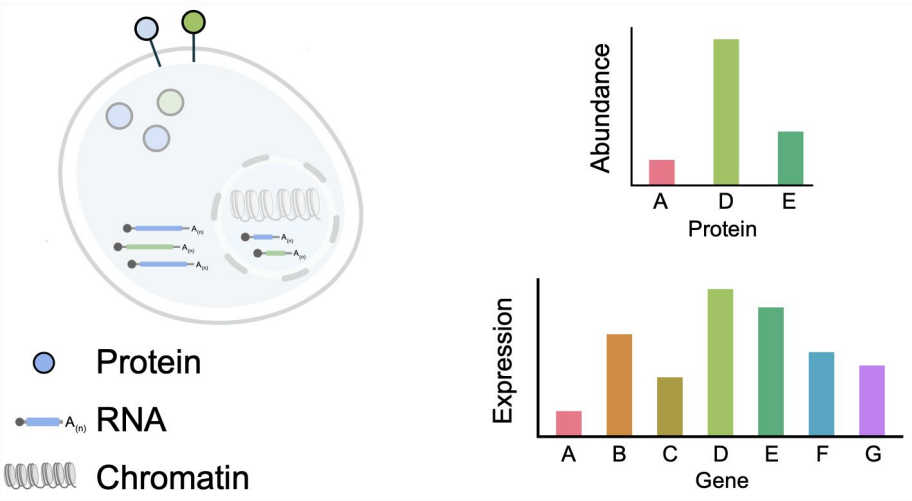
Robust Multi-Omics Prediction (RoMOP) for RNA Expression & Protein Surface Levels

Zachary Abessera & Quentin Chappat

Context



Multimodal scRNA and scATAC from cell nuclei.



Multimodal scRNA and protein abundance from individual cells.

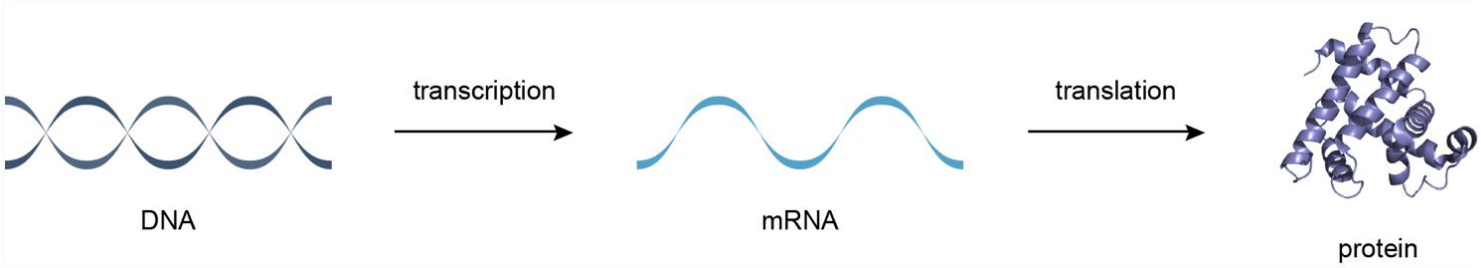
Source: Open Problems in Single-Cell Analysis, *About multimodal single-cell data*

The dataset

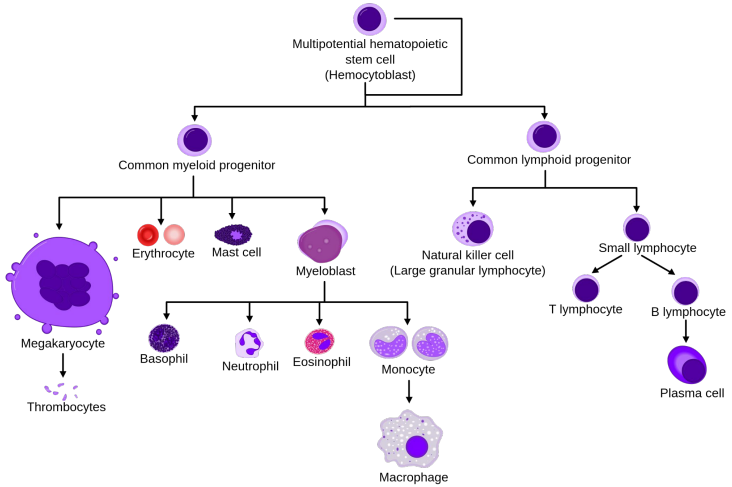
Open Problems in
Single-Cell Analysis

- CD34+ hematopoietic stem and progenitor cells (HSPCs)
- 4 healthy human donors
- 4 time points: days 2, 3, 4, 7
- 2 modalities:
  - **'Multiome'** technology: chromatin accessibility + RNA
  - **'CITESeq'** method: RNA + surface protein level

Our aims



	Train	*Test*
Multiome	days 2,3 & 4	day 7
CITESeq	days 2 & 3	day 4



Source: Wikipedia, *Cellular adoptive immunotherapy*

Implementation: objective functions

Pearson correlation score:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

Mean

Error

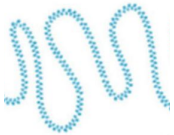
Squared

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Multiome: Pre-processing

Multiome dataset: 105,942 cells x 228,942 genomes x 23,418 RNA gene expression (Total of 90 GB)

Chromatin Accessibility
log(TF)log(IDF)



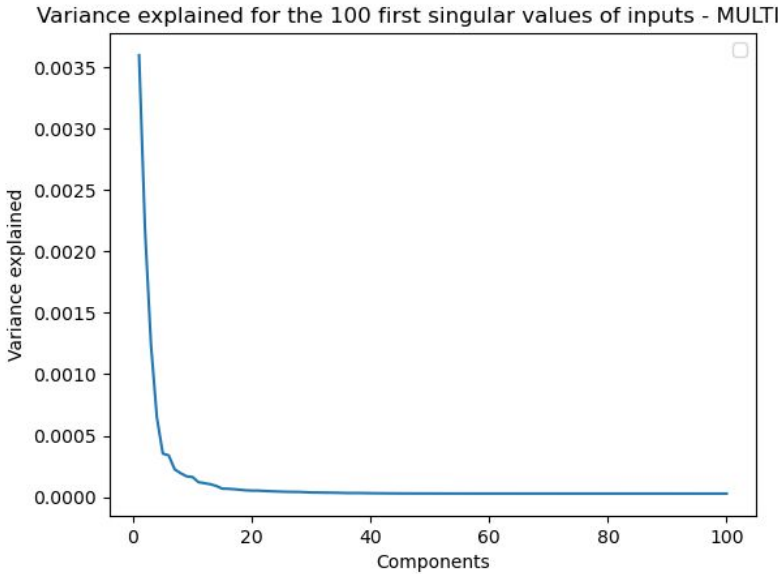
→ Sparse matrix ~ 98% of the row data are 0 → Normalize → Perform SVD to keep and analysis the first 1,000 “singular features”

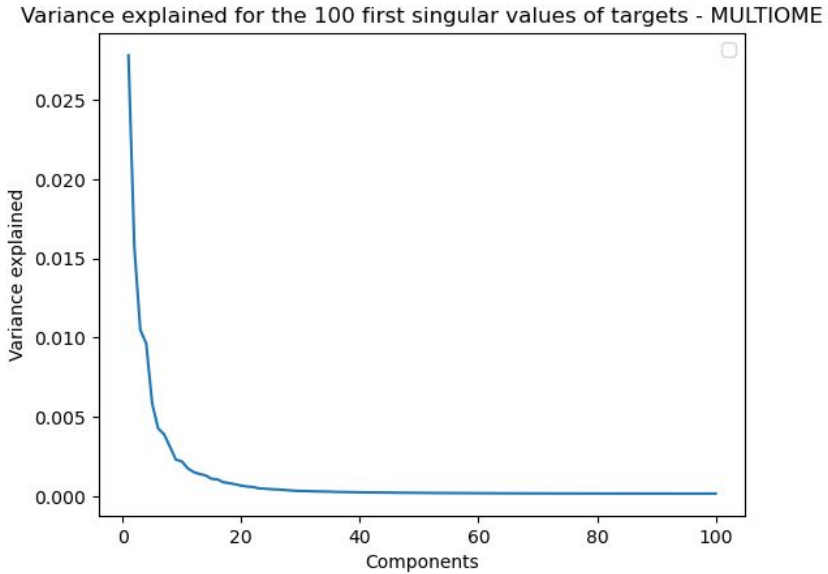
RNA Expression Levels
(Library size normalized & log1p transformed)

→ Sparse matrix → Perform SVD to keep and analysis the first 1,000 “singular targets”

Multiome: Pre-processing

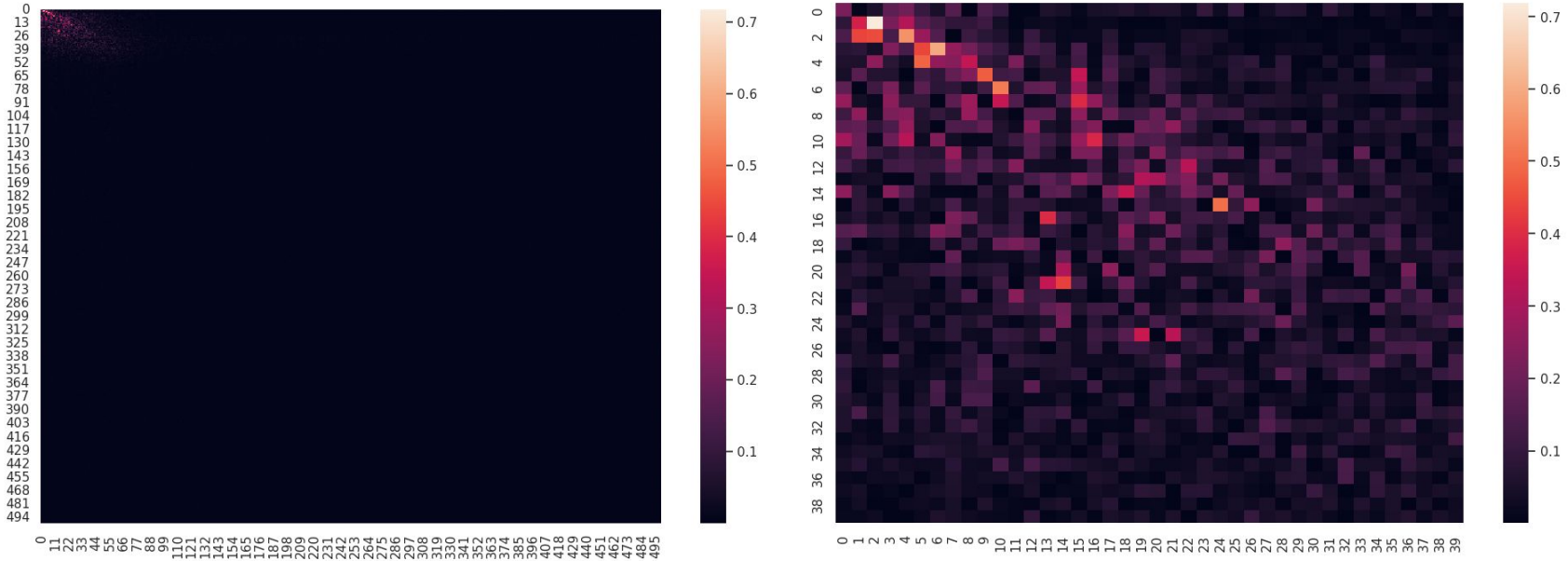
Since our values are mainly 0s, we can consider that all our features' and targets' means are approximately 0 → SVD ~ PCA





Multiome: Pre-processing

Checking that keeping the first 40 singular values for both inputs and targets is correct by computing and plotting the correlations between “singular targets” and “singular inputs”:



Multiome: Pre-processing

Final steps before modeling and training:

→ Keeping for each “singular targets” the 10 most correlated “singular features”
→ + using cell types as dummy variables (cell types were clustered following the methods of ‘Human haematopoietic stem cell lineage commitment is a continuous process’ by Lars Velten et al. 2017)

Recap:

- 40 singular features computed from raw data using SVD
- 7 cell types as dummy variables
- 40 singular targets computed from raw data using SVD
- 4 days & 3 patients = 12 folds

cell_id	day	donor	group	BP	EryP	HSC	MasP	MkP	MoP	...	30_targets	31_targets	32_targets	33_targets	34_targets	35_targets	36_targets	37_targets	38_target
56390cf1b95e	2	32606	2_32606	0	0	0	0	0	0	...	-0.895247	1.950204	0.398738	6.106205	-1.826311	0.203952	3.663262	-2.565385	-0.06280
fc0c60183c33	2	32606	2_32606	0	0	1	0	0	0	...	-4.245243	8.080447	2.193192	6.422708	-0.658925	2.489529	-3.523815	-1.881789	2.34003
9b4a87e22ad0	2	32606	2_32606	0	0	0	1	0	0	...	4.693424	2.098315	-3.740669	2.077569	1.923951	2.322400	-5.129830	-1.301605	0.10061
81cccad8cd81	2	32606	2_32606	0	0	1	0	0	0	...	-2.081603	0.006938	0.018547	-1.178010	1.887824	-7.510479	-1.319409	-1.395976	-3.40567
15cb3d85c232	2	32606	2_32606	0	0	0	0	1	0	...	-1.251107	1.717856	-6.352590	-3.796247	1.079986	0.067629	-1.964458	1.688243	2.05972
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
063cead1a4ea	7	31800	7_31800	0	0	1	0	0	0	...	1.188267	-3.146415	4.226768	0.220194	-1.389867	3.624359	-3.811886	-0.662942	-8.18200
553bca99ba78	7	31800	7_31800	0	0	0	1	0	0	...	-9.933636	-3.626710	-0.160677	2.191283	-0.234477	2.131776	-2.369477	-0.867645	-5.95100
00783f28b463	7	31800	7_31800	0	0	0	0	0	0	...	2.918802	3.407279	-0.568507	-1.637435	-2.053125	5.148231	-8.183878	1.912434	4.72493
e7abb1a0f251	7	31800	7_31800	0	1	0	0	0	0	...	2.254739	-5.538095	-0.057370	4.481005	-2.514055	-2.902745	-4.495811	-6.214432	6.67281
193992d571a5	7	31800	7_31800	0	0	0	1	0	0	...	-6.692030	0.008501	2.088936	-4.527691	-6.127163	11.766088	9.151429	-5.756022	-1.43509

Multiome: Data splitting

Train/test split:

- Making models trained over 1, 2 or all 3 patients
- Always training on days 2, 3 and 4
- Testing on the rest of the data
- Each fold (data/patient/day) is balanced (between 8/9% ~ 1/12 of all data)

Train Dataset		Test Dataset	
Patient(s)	Days	Patient(s)	Day(s)
13176	2, 3 & 4	13176	7
		31800, 32606	2, 3, 4 & 7
31800	2, 3 & 4	31800	7
		13176, 32606	2, 3, 4 & 7
32606	2, 3 & 4	32606	7
		13176, 31800	2, 3, 4 & 7
13176, 31800	2, 3 & 4	13176, 31800	7
		32606	2, 3, 4 & 7
31800, 32606	2, 3 & 4	31800, 32606	7
		13176	2, 3, 4 & 7
13176, 32606	2, 3 & 4	13176, 32606	7
		31800	2, 3, 4 & 7
13176, 31800, 32606	2, 3 & 4	13176, 31800, 32606	7

Multiome: Modeling

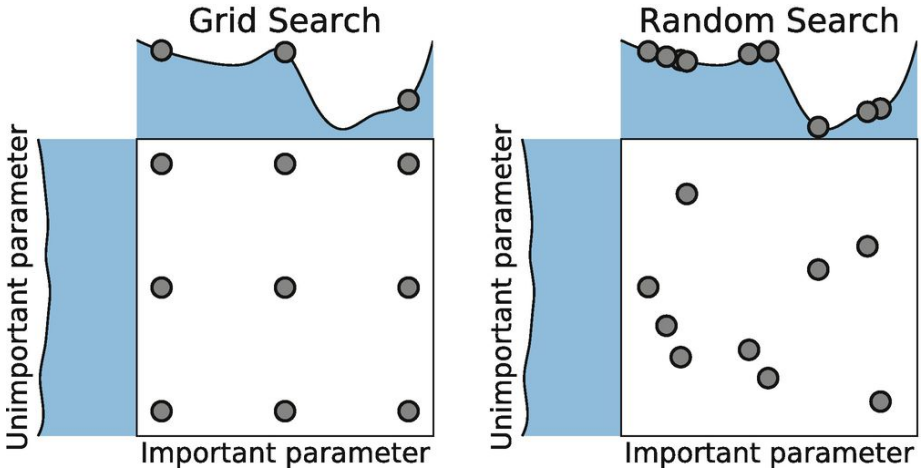
In “Integrative prediction of gene expression with chromatin accessibility and conformation data” by Florian Schmidt et al. from 2020, the authors used ElasticNet (mix of Ridge & Lasso) to predict gene expression from chromatin accessibility:

$$\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2$$

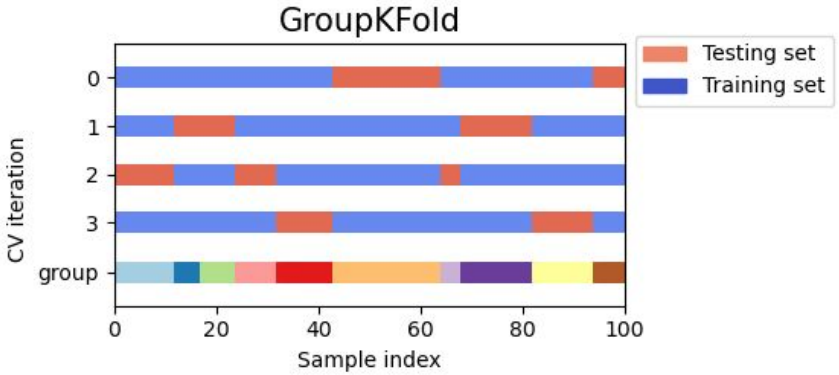
Therefore, we decided to train the followings models using scikit-learn framework:

- Elastic Net
- Linear Regression $\min_w ||Xw - y||_2^2$
- Lasso $\min_w ||Xw - y||_2^2 + \alpha||w||_2^2$
- Ridge Regression $\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha||w||_1$
- Bayesian Ridge Regression $p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}\mathbf{I}_p)$
- Automatic Relevance Determination $p(w|\lambda) = \mathcal{N}(w|0, A^{-1})$ & $A = \text{diag}(\{\lambda_1, \ldots, \lambda_p\})$

Multiome: Training



Used RandomSearchCV to validate the models and optimize the hyperparameters of each models



Used GroupKFold with CV where each fold is the data of one day of one patient

Source: Scikit-learn documentation, Cross-validation

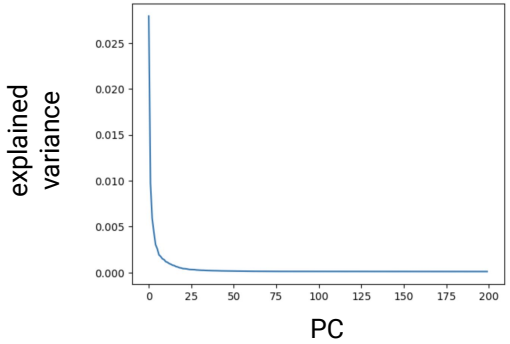
Multiome: Results

Train Dataset	Model	MSE (Train)	Pearson (Train)	MSE (Test)	Pearson (Test)
13176	Linear Regression	2.008	0.677	2.063	0.653
13176	Maximum	1.920	0.689	1.927	0.674
31800	Linear Regression	2.037	0.673	2.048	0.655
31800	Maximum	1.933	0.687	1.922	0.675
32606	Linear Regression	2.007	0.678	2.061	0.654
32606	Maximum	1.921	0.690	1.926	0.674
13176, 31800	Linear Regression	2.025	0.675	2.063	0.644
13176, 31800	Maximum	1.926	0.688	1.923	0.667
31800, 32606	Linear Regression	2.024	0.675	2.064	0.644
31800, 32606	Maximum	1.927	0.688	1.923	0.667
13176, 32606	Linear Regression	2.009	0.678	2.078	0.643
13176, 32606	Maximum	1.920	0.690	1.930	0.666
13176, 31800, 32606	Linear Regression	2.02	0.676	2.121	0.605
13176, 31800, 32606	Maximum	1.925	0.689	1.926	0.638

CITESeq: gene expression -> protein

- Input: gene expression level (RNA library-size normalized and log1p transformed counts)
- Output: surface protein level (dsb* normalized)
- 70,988 cells x 22,050 genes
- Baseline model:
  - Dimensionality reduction: PCA (n = 50)
  - Multi-Output Linear Regression
  - Train on days 2 & 3, Test on day 4

*denoised and scaled by background



In [8]:
```
meta_day23 = metadata_citeseq[(metadata_citeseq.day == 2) | (metadata_citeseq.day == 3)]
meta_day23.groupby('day').count()
```

Out[8]:

day	cell_id
2	29418
3	27389

In [10]:
```
train_inputs = cite_inputs[cite_inputs.index.isin(meta_day23.cell_id)]
train_inputs.shape
```

Out[10]:
```
(42843, 22050)
```

In [12]:
```
test_inputs = cite_inputs[cite_inputs.index.isin(meta_day4.cell_id)]
test_inputs.shape
```

Out[12]:
```
(28145, 22050)
```

	Linear Reg. with PCA
Correl.	0.74
MSE	6.88

CITESeq: gene expression -> protein

- Preprocessing:
  - *constant_cols*: constant features are discarded.
  - *important_cols*: all features whose name matches the name of a target protein. They don't undergo dimensionality reduction.
    Example: gene 'ENSG00000114013_CD86' as an input, should be related to protein 'CD86'
- Convert to sparse matrix
- Dimensionality reduction: truncated SVD (n = 512), can work with sparse matrices efficiently.

	Linear Reg. with PCA	Linear Reg. with Truncated SVD and preprocessing
Correl.	0.74	0.88
MSE	6.88	2.74

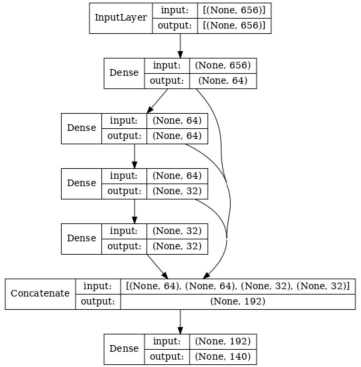
CITESeq: gene expression -> protein

- New model: LightGBM
  - gradient boosting framework that uses tree based learning algorithm
  - fast and designed to handle large data sets
  - memory-efficient

	Linear Reg. with PCA	Linear Reg. with Truncated SVD and preprocessing	LightGBM with PCA	LightGBM with Truncated SVD and preprocessing
Correl.	0.74	0.88	0.76	0.89
MSE	6.88	2.74	6.21	2.70

CITESeq: gene expression -> protein

- New model: sequential dense network with four hidden layers
  - hyperparameters tuning with Bayesian Optimization tuner from Keras
    - sizes of the hidden layers
    - regularization factors



	Linear Reg. with PCA	Linear Reg. with Truncated SVD and preprocessing	LightGBM with PCA	LightGBM with Truncated SVD and preprocessing	Neural Net (Bayesian Optimisation), Truncated SVD and preprocessing
Correl.	0.74	0.88	0.76	0.89	0.90
MSE	6.88	2.74	6.21	2.70	

Conclusion

- Learned how to handle big data (preprocessing, sparse matrices, truncated SVD)
- Performed hyperparameters search using bayesian optimization & cross-validation
- Built a robust framework of predictions over patient and days

- Difficulties to make biological interpretations of our features because of SVD
- Showed that most of the information is concentrated and enable us to obtain good results

- Future work would be to analyse the trade-off during SVD between losing information and being able to handle the amount of data & work on the interpretation of the results to make biological analysis

Thank you!