Final Report - Using deep autoencoder feature embeddings to explore single-cell phenotypes in pediatric cancer

Timothy Keyes

Abstract

Single-cell data are collected in both research and clinical laboratories in order to evaluate leukemia patients' blood samples for the presence of various cell types. Specifically, the enumeration of various developmental "blast" types within leukemic samples is often of clinical and research interest. However, state-of-the-art methods for measuring cells' biological characteristics are prone to batch effects and other forms of noise due to differences in individual laboratories' instrumentation, operators, and on-hand reagents. Here, we adapt a previously-published, multi-tasking autoencoder called "SAUCIE" (Sparse Autoencoder for Unsupervised Clustering, Imputation, and Embedding) to denoise, cluster, and visualize single-cell data from human blood samples taken from B-cell Precursor Acute Lymphoblastic Leukemia (BCP-ALL) patients. We evaluate the performance of the algorithm compared to a gold-standard, supervised clustering algorithm developed specifically for BCP-ALL data and investigate its population-specific performance.

1 - Introduction

Problem importance and motivation

In the clinical evaluation of leukemia (blood cancer), most diagnostic and prognostic tests rely on the identification and enumeration of leukemic "blasts" in the blood and bone marrow of patients. Blasts are immature blood cells that - due to genetic and epigenetic abnormalities - develop aberrancies in cellular maturation that cause them to become cancerous. Blast phenotypes differ widely between patients both because of individual differences in the biology of each patient's cancer and because of instrumentation differences between clinics where testing is conducted. This means that the current gold standard of diagnostic and prognostic testing for leukemia relies on pathologists manually inspecting the protein-level phenotypes of cancer patients by eye using microscopy,[1] and flow cytometry.[2]

Here we use deep learning to take a step towards automating methods of enumerating leukemic blasts by building an autoencoder framework capable of denoising, batch-correcting, and clustering protein-level data collected by single-cell cytometry such that individual differences in protein marker expression are preserved but instrument-to-instrument differences are reduced. We also use this multi-tasking autoencoder to extract clustering information from one layer of the network for the unbiased indentification of blast-like cells within the denoised feature space.

Input and Output

The input for this algorithm is a `.fcs` file[3] (or folder of `.fcs` files) containing single-cell information collected via a flow cytometer or a mass cytometer, two types of instrument that collect protein-level features from individual cells.[2] The data represented in an .fcs file can be represented as an [m x n] matrix in which `m` represents the number of cells that you've measured and `n` represents the number of proteins that you've measured within each cell.

The output of the algorithm includes the following:

1. A reconstructed `n`-dimensional feature-vector for each cell after being passed through the autoencoder.
2. A label for each cell indicating which cluster it was assigned to by the decoder of the network (see methods below). The clusters are identified in an unsupervised manner and the number of clusters is automatically detected by the network.

2 - Related Work

Only a small amount of previous work has applyied deep learning to single-cell data, and an especially small amount of work has applied deep learning to flow or mass cytometry data. Specifically, 3 algorithms have been previously published that apply deep learning to single-cell cytometry data: DeepCyTOF,[4] CellCNN,[5] and SAUCIE.[6] Details about each of these algorithms are provided below.

DeepCyTOF is a feed-forward neural network that automates the task of "gating" immune cell populations, a process whereby immunologists identify cell populations of interest manually by sequentially applying two-dimensional, hand-drawn filters to single-cell data.[4] DeepCyTOF was the first deep learning algorithm applied to mass cytometry data, and was built as a depth-4 fully-connected network with softplus hidden units (except for the output layer, which was a softmax layer). Layer sizes were 12, 6, 3, and 1, respectively and were trained with RMSprop on a training set of blood samples taken from 14 patients infected with West Nile Virus. Importantly, DeepCyTOF was developed in order to identify distinct immune cell populations (which are known to differ widely from one another) and has not been tested on leukemic cell types.

CellCNN (which stands for "Cellular Convolutional Neural Network") is a convolutional neural network approach to identifying cellular subtypes within a sample that associate with a particular clinical outcome of interest.[5] In short, CellCNN works by applying 1-3 m-dimensional filters to each cell in the training set (where `m` is the number of proteins measured in each cell). The convolutional layer is followed by either a max pooling or average pooling layer, allowing the algorithm to identify the 1-3 major phenotypes associated with the clinical outcome on a single-cell level. The strengths of CellCNN include that it is able to identify relatively rare cell types that associate with clinical outcomes while ignoring cell types that aren't associated with the outcome-of-interest. However, a weakness of CellCNN is that it doesn't perform any denoising or batch correction, which prevents it from being a fully end-to-end approach even when samples are clinically annotated.

This project leans heavily on existing implementations of a network method called *SAUCIE* developed using non-cancer cells and that we have adapted for this project.[6] SAUCIE ("sparse autoencoder for unsupervised clustering, imputation, and embedding") is a multitasking autoencoder that "extends" the architecture of a simple autoencoder. It does so by adding several layers that are regularized in such a way that their outputs are biologically meaningful. These layers include a clustering layer that is regularized to penalize within-cluster distances between cells *and* that performs information dimension (ID) regularization (to encourage cluster sparsity); in addition, the "embedding" layer of the autoencoder (i.e. the middle layer between the encoder and the decoder) is regularized such that the pairwise distances between cells from different batches is minimized, allowing for denoising and batch correction. Details of the SAUCIE network are described below.

3 - Dataset and Features

In this project, I am working with data originally published in a previous paper from my lab.[7] The paper devised a method of "aligning" cancer cells with the healthy cell type with which they are most similar. Overall, the main idea of this paper was that comparing cancer cell subtypes to healthy cell subtypes might help us to infer how cancer cells behave or where they come from, and its main contribution to the field was a gold-standard clustering algorithm (called the "developmental classifier") that serves as a gold-standard method of annotating cancer cell types into their most biologically-interpretable cellular subpopulation. Specifically, the dataset obtained during this study included mass cytometry data from 60 patients with B-cell precursor acute lymphoblastic leukemia (BCP-ALL) and 5 healthy control patients.[8]

Basic dataset characteristics

For this project, I worked with a cleaned version of these data to eliminate some of the frustration of working with samples that vary significantly in size and quality. Specifically, I limited the dataset to solely diagnostic specimens taken from the blood or bone marrow, and I sampled 10,000 cells from each unique cell subpopulation identified by the developmental classification algorithm described above. All patients that did not have at least 10,000 cells total in their sample were removed from the analysis. Information regarding the number of cells and basic summary statistics of the dataset are provided in the Appendix (*Table 1* and *Table 2*).

Pre-processing

Data pre-processing was performed as is standard for CyTOF data analysis.[9] Specifically, the following steps were performed: 1) All measurement values ("ion counts") taken from the cytometry were arsinh-transformed with a cofactor of 5 such that $final\ value = arcsinh(\frac{counts}{5})$. 2) Mass cytometers are prone to artefacts on both the upper- and lower- ends of their dynamic range. Thus, protein measurement values at or below the 1st percentile and at or above the 99th percentile of the measured ranges were excluded as outliers. 3) Protein expression values for remaining cells were centered and scaled such that each protein (each feature) had a mean of 0 and standard deviation of 1.

Example inputs

A single-cell feature vector in the input will look like this:

CD19	CD20	CD34	CD38	IgMi	IgMs	CD179a	CD179b	CD127	Tdt	CD45	PLCg2	CD22	p4EBP1
-0.5	-0.31	-1.12	-0.15	0.19	-0.68	1.18	0.24	-0.62	-0.91	-0.91	1.55	0.36	-0.72

Each cell is associated with 36 protein markers, each of which will provide 1 value in the cell's feature vector. Likewise, each cell population (of which there are 15 according to the gold-standard), will be associated with a distribution of marker expression levels for each protein. An example of what the distributions for two cell populations looks like relative to one another is provided in the Appendix (*Figure S2*).

4 - Methods

Description of Network Architecture

SAUCIE is a multitasking autoencoder that has 3 encoding layers, 1 embedding layer, and three decoding layers, all of which are fully-connected. The number of neurons per hidden layer in the encoder were 512, 256, and 128 with a symmetric decoder; the embedding layer had 2 neurons. The embedding layer used a linear activation function; all other layers used a leaky rectified linear unit activation with leak = 0.2 function except the final layer of the decoder (the layer responsible for performing clustering), which used a rectified linear unit (without leak). Batches of size 300 were used for 1000 steps (in order to train on the entire dataset for 20 epochs). The optimization was performed with ADAM and a learning rate of 0.001.

Loss function

Our adaptation of SAUCIE has 3 components to its loss function. The first component is the simple reconstruction penalty present in any autoencoder, for which we used the mean-squared error of the output representation and the original input data. If we consider the following quantities...

- x_i = the input feature vector of the ith cell in the dataset
- $\hat{x}_i$ = the output feature vector of the ith cell in the dataset
- m = the number of cells in the input dataset,

Then the loss term from the reconstruction is given by the following:

$$L_{reconstruction} = \frac{1}{m}\sum_{i}^{m}(x_i - \hat{x}_i)^2$$

The other two components of the loss function are derived from the "clustering" layer (the second-to-last decoding layer), which performs two (opposing) kinds of regularization. The first regularization penality in this layer is an interformation dimension (ID) regularization that encourages activations of the neurons in the layer to be binarized such that sparser representations (i.e. representations with fewer clusters) are penalized less. Thus, the ID regularization term (applied to the clustering layer) is given by

$$L_{ID} = -\sum_{i=1}^{k} p_i log(p_i)$$

where k is the number of neurons in the clustering layer and p_i normalized activation of the ith neuron in the clustering layer a_i such that $p_i = \frac{a_i}{||a_i||_1}$.

Finally, the last component of the loss function is a penalty on intracluster distances for the clusters in the clustering layer of the network, which is calculated as the Euclidian distance between points that are located in the same cluster (in the denoised output space):

$$L_{cluster\ distances} = \sum_{i,j} ||\hat{x}_i - \hat{x}_j||^2$$

for all cells i, j assigned to the same cluster by the network. Thus, while L_{ID} is minimized by assigning all cells to the same cluster, $L_{cluster\ distances}$ acts as an opposing balance, as it will be minimized only if each cell were placed in a cluster by itself.

Thus, the total loss function is given by $L_{total} = L_{reconstruction} + \lambda_c * L_{ID} + \lambda_d * L_{cluster\ distances}$, where λ_c and λ_d are tuning parameters that weight the degree that the ID regularization and the intracluster distance penalty will affect the clusteringl layer. In our implementation, we tuned these hyperparameters by performing a grid search over the values 0-1 at 0.1 intervals and selecting the combination with optimal performance.

5 - Experiments/Results/Discussion

Hyperparameter tuning

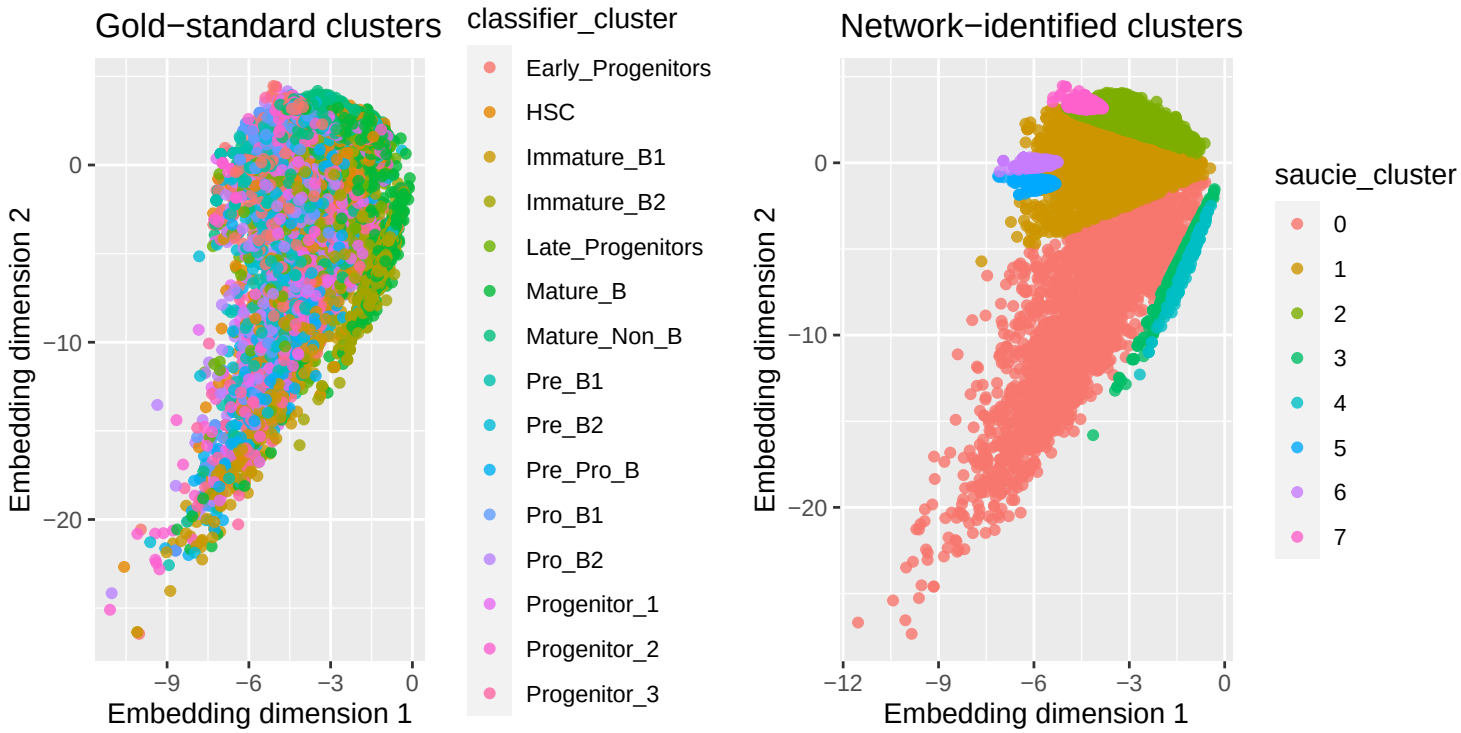
Default values for the number of neurons to use in each layer were used, by the hyperparameters λ_c and λ_d were tuned by calculating each result's performance using a previously-validated metric for analyzing clustering results with single-cell cytometry data.[10] Specifically, because SAUCIE is an unsupervised learning algorithm, we evaluated its performance by comparing it to the "gold-standard" supervised clustering algorithm that the authors applied to the same data in the original paper for which the data were collected. Specifically, we compared SAUCIE's performance to the original authors' algorithm using a version of the F1-measure of classification accuracy commonly used to compare single-cell clustering methods to one another. In short, the F1-measure is the harmonic mean of precision and recall for classification compared to a gold-standard method. SAUCIE performed with an F1-measure overall of 0.19, making it slightly less than average as far as clustering algorithms applied to mass cytometry datasets are concerned.

Population-specific performance is shown here:

Developmental Population	Precision	Recall	F-measure
Mature_Non_B	0.3023151	0.6066280	0.4035298
Mature_B	0.1953897	0.3941621	0.2612670
Immature_B1	0.1274997	0.9846954	0.2257669
Early_Progenitors	0.1198223	0.5873143	0.1990375
Pre_Pro_B	0.1096778	0.8470541	0.1942090
Progenitor_3	0.1070355	0.8266480	0.1895304
Pro_B1	0.1072504	0.6226079	0.1829805
Pre_B1	0.0956119	0.7482262	0.1695569
Late_Progenitors	0.1015377	0.4964958	0.1685961
Pre_B2	0.0921190	0.4503504	0.1529518
Progenitor_1	0.0812222	0.6276021	0.1438303
Pro_B2	0.0802767	0.6211043	0.1421772
Progenitor_2	0.0783186	0.3828829	0.1300379
Immature_B2	0.0711584	0.5516618	0.1260569
HSC	0.0755134	0.3690214	0.1253718

Embedding

If we look at the values for each cell in the embedding dimension of the autoencoder, we can see that the embedding doesn't seem to partition the gold-standard clusters from each other very well, but it does partition the clusters identified by the network itself quite well...



These plots suggest that one way to improve the performance of the unsupervised clustering might be to include more neurons in the embedding dimension (as whatever mapping is being performed by the gold-standard categorizations may not be captured well in just 2 dimensions).

6 - Conclusions/Future Work

In general, our algorithm was not particularly well-performing compared to the gold-standard. This is probably because the clusters that SAUCIE learns to identify are the ones that allow the best reconstruction of the input data (in our case, the input 36-dimensions) and not necessarily the clusters with the most biological interpretability (which is what the gold-standard algorithm is meant to represent). This may indicate that that a purely unsupervised approch to cluster identification may not be the best strategy for detecting rare cell types of a particular developmental origin.

In future iterations of this project, I would be interested in combining the autoencoder that I experimented with here and some of the supervised approaches that have been used in either CellCNN (to incorporate a component of the loss function that is associated with a clinical outcome) or DeepCyTOF (to incorporate a component of the loss function based on having direct access to the class label for each cell).

7 - Contributions

I was the only team member for this project, so I performed all analyses myself.

References

1. Abou Dalle I, Jabbour E, Short NJ. Evaluation and management of measurable residual disease in acute lymphoblastic leukemia. Ther Adv Hematol. 2020;11:2040620720910023. Published 2020 Mar 6. doi: 10.1177/2040620720910023
2. Wang XM. Advances and issues in flow cytometric detection of immunophenotypic changes and genomic rearrangements in acute pediatric leukemia. Transl Pediatr. 2014;3(2):149-155. doi:10.3978/j.issn.2224-4336.2014.03.06
3. https://en.wikipedia.org/wiki/Flow_Cytometry_Standard

4. Li H, Shaham U, Stanton KP, Yao Y, Montgomery RR, Kluger Y. Gating mass cytometry data by deep learning. Bioinformatics. 2017;33(21):3423-3430. doi:10.1093/bioinformatics/btx448

5. Arvaniti, E., Claassen, M. Sensitive detection of rare disease-associated cell subsets via representation learning. Nat Commun 8, 14825 (2017). https://doi.org/10.1038/ncomms14825

6. Amodio, M., van Dijk, D., Srinivasan, K. et al. Exploring single-cell data with deep multitasking neural networks. Nat Methods 16, 1139–1145 (2019). https://doi.org/10.1038/s41592-019-0576-7

7. Good Z, Sarno J, Jager A, et al. Single-cell developmental classification of B cell precursor acute lymphoblastic leukemia at diagnosis reveals predictors of relapse. Nat Med. 2018;24(4):474-483. doi:10.1038/nm.4505

8. Finck R, Simonds EF, Jager A, et al. Normalization of mass cytometry data with bead standards. Cytometry A. 2013;83(5):483-494. doi:10.1002/cyto.a.22271

9. Olsen LR, Leipold MD, Pedersen CB, Maecker HT. The anatomy of single cell mass cytometry data. Cytometry A. 2019;95(2):156-172. doi:10.1002/cyto.a.23621

10. Aghaeepour, N., Finak, G., Hoos, H. et al. Critical assessment of automated flow cytometry data analysis techniques. Nat Methods 10, 228–238 (2013). https://doi.org/10.1038/nmeth.2365

Appendix

Table 1 - Summary table of patient cell counts

patient	Number of cells
Healthy1	194185
Healthy2	1003294
Healthy3	2552004
Healthy4	662174
Healthy5	81935
UPN1	857516
UPN1-Relapse	10330
UPN10	62436
UPN10-Relapse	158774
UPN11	353739
UPN12	856763
UPN13	356801
UPN14	357327
UPN15	720760
UPN16	471291
UPN17	1054732
UPN18	216344
UPN19	701822
UPN2	310516
UPN20	640674
UPN21	162456
UPN22	34637
UPN22-Relapse	42027
UPN23	508632
UPN24	481026
UPN25	768576
UPN26	351252
UPN27	782173
UPN28	340810
UPN29	722647
UPN3	161258
UPN30	765328
UPN31	377982
UPN35	3525
UPN35-Relapse	14228
UPN4	622846
UPN45	138423
UPN45-Relapse	61082
UPN47	255890
UPN48	306940
UPN49	468008
UPN5	659056
UPN50	388890
UPN51	298401
UPN52	370408
UPN53	564090
UPN54	545150
UPN55	500697
UPN56	401562
UPN57	134172
UPN58	226294
UPN6	752164
UPN60	121538
UPN60-Blood	197192
UPN61-Blood	111542
UPN62	243358
UPN62-Blood	61112
UPN63	214145
UPN63-Blood	51455
UPN64-Blood	12919
UPN65-Blood	50700
UPN67	62025
UPN68	79601
UPN69	227090
UPN7	908214
UPN8	730373
UPN9	870053
UPN90	447251

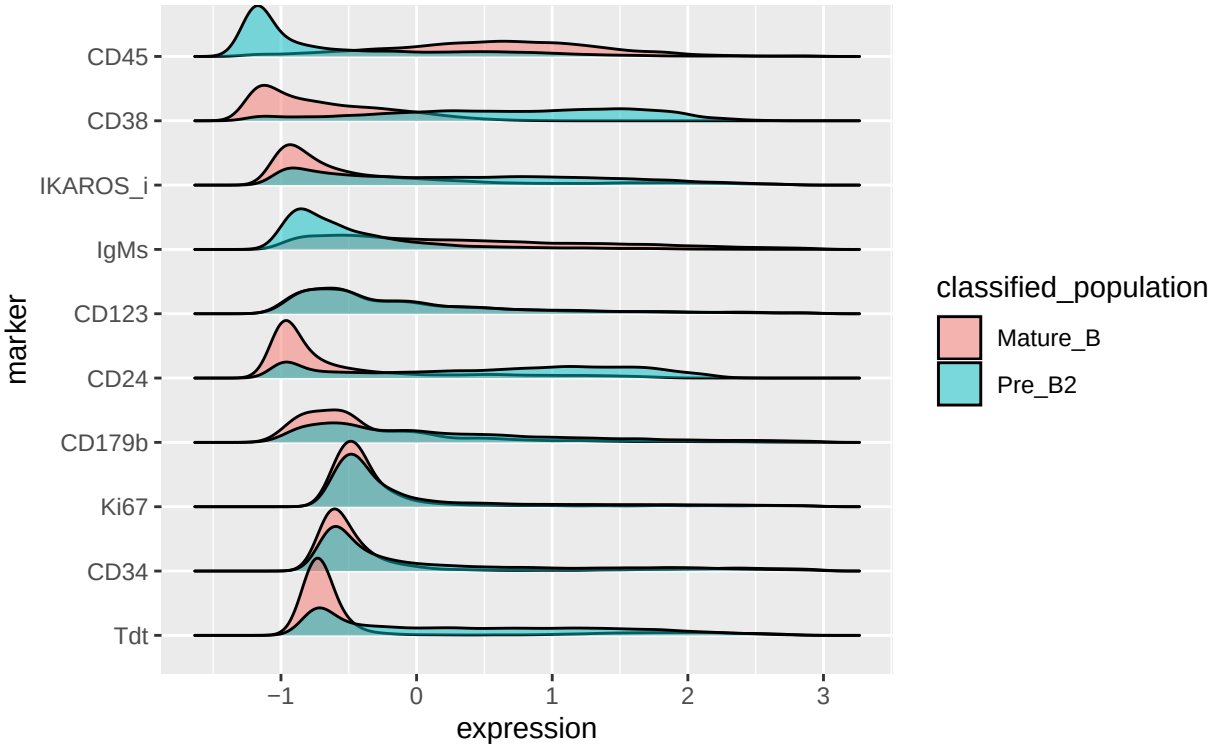
Table 2 - Summary statistics for the dataset

```
##    patient              CD45               PLCg2              CD19
##  Length:115879      Min.   :  -1.1313   Min.   :-1.1343   Min.   : -1.122
##  Class :character   1st Qu.:  -0.3058   1st Qu.:-0.7013   1st Qu.:  2.981
##  Mode  :character   Median :   0.7924   Median :-0.4202   Median :  9.091
##                     Mean   :   2.3708   Mean   :-0.3214   Mean   : 15.668
##                     3rd Qu.:   3.0605   3rd Qu.:-0.1413   3rd Qu.: 20.814
##                     Max.   :1617.0559   Max.   :82.6025   Max.   :399.320
##       CD22              p4EBP1             Ikaros            CD79b
##  Min.   :  -1.121   Min.   :-1.1270   Min.   :-1.113   Min.   :  -1.1276
##  1st Qu.:  -0.385   1st Qu.:-0.3457   1st Qu.: 1.041   1st Qu.:  -0.4054
##  Median :   0.344   Median : 0.4785   Median : 3.280   Median :   0.2837
##  Mean   :   1.251   Mean   : 1.2535   Mean   : 4.768   Mean   :   0.8862
##  3rd Qu.:   1.679   3rd Qu.: 1.9454   3rd Qu.: 6.852   3rd Qu.:   1.4534
##  Max.   :3947.908   Max.   :36.0411   Max.   :85.211   Max.   :1238.4268
##       CD20               CD34              CD179a              pSTAT5
##  Min.   :  -1.1232   Min.   :  -1.108   Min.   :  -1.1302   Min.   :  -1.129
##  1st Qu.:   0.1103   1st Qu.:  11.049   1st Qu.:  -0.6377   1st Qu.:  -0.580
##  Median :   2.7838   Median :  32.155   Median :  -0.2929   Median :  -0.177
##  Mean   :  11.0486   Mean   :  48.279   Mean   :  -0.0403   Mean   :   0.286
##  3rd Qu.:  11.5035   3rd Qu.:  70.023   3rd Qu.:   0.1788   3rd Qu.:   0.592
##  Max.   :2362.1838   Max.   :2956.034   Max.   :2791.2075   Max.   :4353.186
##      CD123               Ki67               IgMi            IgL kappa
##  Min.   : -1.1136   Min.   :  -1.120   Min.   :  -1.131   Min.   :  -1.1264
##  1st Qu.: -0.2473   1st Qu.:  -0.319   1st Qu.:  -0.548   1st Qu.:  -0.5782
##  Median :  0.7159   Median :   0.855   Median :  -0.110   Median :  -0.1744
##  Mean   :  1.6628   Mean   :   6.304   Mean   :   1.494   Mean   :   0.2510
##  3rd Qu.:  2.4637   3rd Qu.:   5.311   3rd Qu.:   0.755   3rd Qu.:   0.5888
##  Max.   :256.0133   Max.   :7137.480   Max.   :6827.284   Max.   :1119.9121
##     IKAROS_i              CD10              CD179b              pAkt
##  Min.   : -1.11967   Min.   :  -1.076   Min.   :-1.1307   Min.   :-1.12993
##  1st Qu.: -0.00687   1st Qu.: 265.449   1st Qu.:-0.3599   1st Qu.:-0.65985
##  Median :  1.58254   Median : 412.521   Median : 0.3895   Median :-0.33451
##  Mean   :  8.69393   Mean   : 448.916   Mean   : 0.9981   Mean   :-0.10036
##  3rd Qu.:  9.76113   3rd Qu.: 592.952   3rd Qu.: 1.6056   3rd Qu.:-0.00698
##  Max.   :306.35855   Max.   :8207.047   Max.   :79.2710   Max.   :96.83366
##       CD24               CRLF2              CD127               RAG1
##  Min.   :   -1.089   Min.   :  -1.1240   Min.   :  -1.1319   Min.   :  -1.124
##  1st Qu.:  122.943   1st Qu.:  -0.4865   1st Qu.:  -0.6057   1st Qu.:  -0.635
##  Median :  249.645   Median :   0.0086   Median :  -0.2253   Median :  -0.288
##  Mean   :  354.999   Mean   :   0.5176   Mean   :   0.1687   Mean   :   0.257
##  3rd Qu.:  472.004   3rd Qu.:   0.9129   3rd Qu.:   0.4486   3rd Qu.:   0.240
##  Max.   :11929.121   Max.   :1926.6940   Max.   :1422.5455   Max.   :3502.190
##       Tdt                Pax5               pSyk                CD43
##  Min.   :  -1.121   Min.   : -1.120   Min.   : -1.12314   Min.   :  -1.104
##  1st Qu.:   0.368   1st Qu.:  4.077   1st Qu.: -0.61694   1st Qu.:  13.060
##  Median :   1.903   Median : 11.326   Median : -0.24565   Median :  37.675
##  Mean   :   3.327   Mean   : 16.081   Mean   :  0.06261   Mean   :  68.454
##  3rd Qu.:   4.642   3rd Qu.: 22.791   3rd Qu.:  0.38189   3rd Qu.:  86.343
##  Max.   :4339.835   Max.   :266.003   Max.   :141.63513   Max.   :8124.547
##       CD38               CD58              HIT3a               CD16
##  Min.   :  -1.1217   Min.   :  -1.112   Min.   :-1.13152   Min.   :-1.1275
##  1st Qu.:   0.5536   1st Qu.:   1.588   1st Qu.:-0.66915   1st Qu.:-0.4244
```

```
##  Median :   2.9406   Median :   4.601   Median :-0.35835   Median : 0.2393
##  Mean   :   9.7400   Mean   :   7.016   Mean   :-0.19297   Mean   : 0.8588
##  3rd Qu.:   9.2547   3rd Qu.:   9.658   3rd Qu.:-0.04804   3rd Qu.: 1.3881
##  Max.   :3106.6267   Max.   :2605.045   Max.   : 8.32404   Max.   :34.1563
##       pS6                pErk              HLADR               IgMs
##  Min.   :  -1.133   Min.   : -1.1196   Min.   :  -1.066   Min.   :  -1.1304
##  1st Qu.:   0.957   1st Qu.: -0.5621   1st Qu.:  68.323   1st Qu.:  -0.4188
##  Median :   3.473   Median : -0.1369   Median : 162.820   Median :   0.2217
##  Mean   :   9.350   Mean   :  0.2829   Mean   : 270.851   Mean   :   0.9521
##  3rd Qu.:   8.272   3rd Qu.:  0.6816   3rd Qu.: 354.250   3rd Qu.:   1.3034
##  Max.   :6388.095   Max.   :104.1688   Max.   :7318.328   Max.   :2265.0278
##      pCreb
##  Min.   :  -1.111
##  1st Qu.:   1.578
##  Median :   5.202
##  Mean   :   9.686
##  3rd Qu.:  12.550
##  Max.   :3152.514
```

Most important to note here is that, as is common with mass cytometry data (particularly in cancer), the distributions are highly skewed such that there are often huge(!) outliers in the positive direction due to instrumentation failure. These values are not biologically informative, so filtering out all measurements that are above the 95th percentile in a given channel was performed.

Figure S1 - Example marker distributions



Density plots for two gold-standard cell populations are provided above for several example protein markers in the dataset. Notably, features can differ between cell populations both in their central tendency and dispersion, and some markers will be more/less similar than others across multiple cell populations.