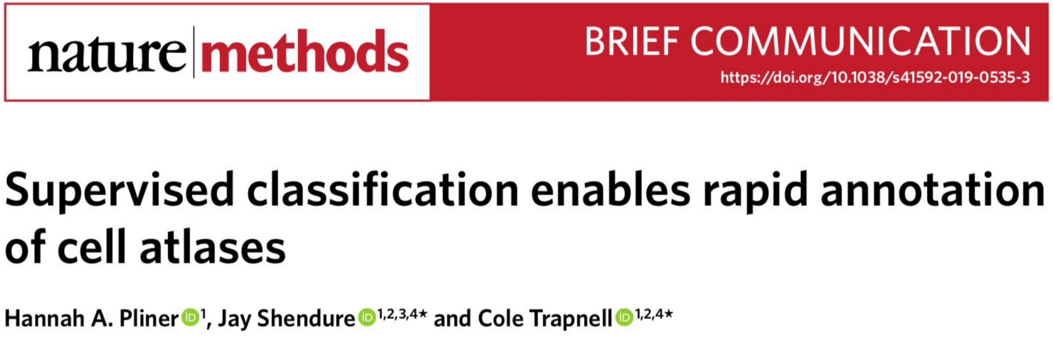
Garnett:
Supervised cell type classification

Luis Haddock-Soto, Matthew Stone
BME 780 - 16 October 2019



nature methods

BRIEF COMMUNICATION

https://doi.org/10.1038/s41592-019-0535-3

Supervised classification enables rapid annotation of cell atlases

Hannah A. Pliner[1], Jay Shendure[1,2,3,4*] and Cole Trapnell[1,2,4*]

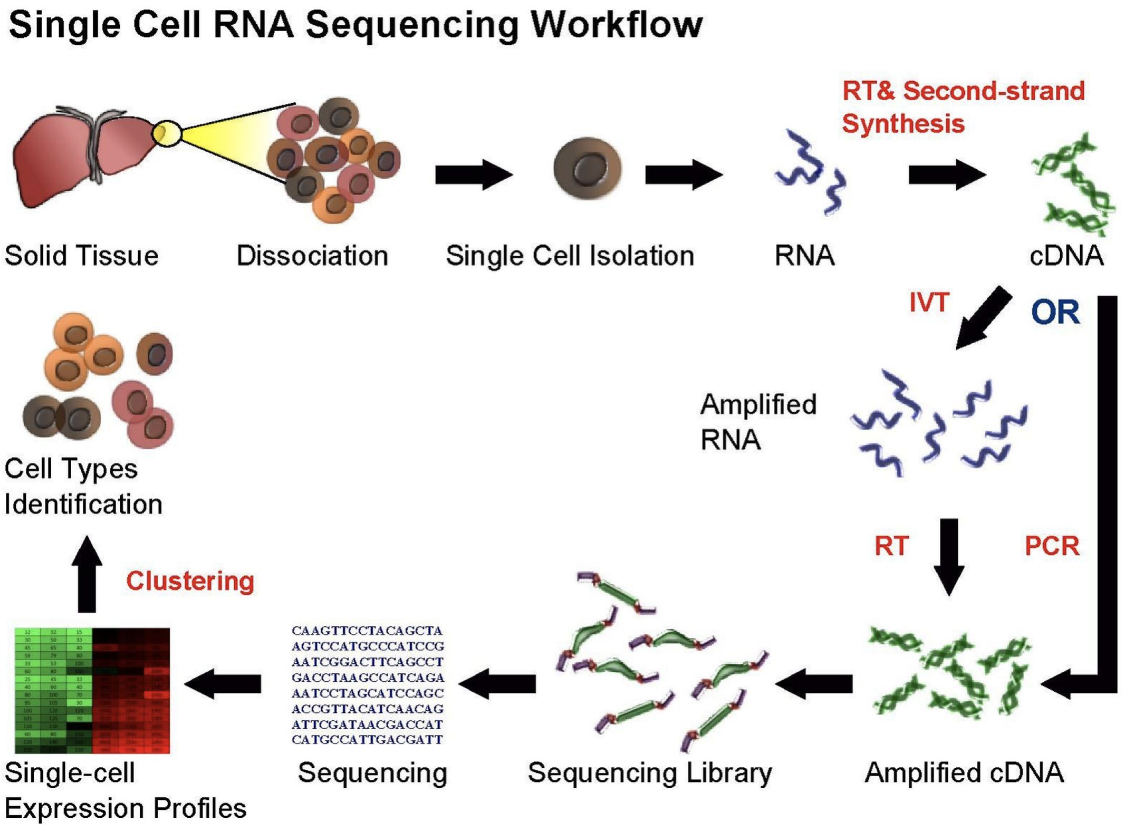
Biological context:

Single-cell RNA-sequencing and cell type annotation

Single-cell RNA-seq

- By analyzing **transcriptomes**, we can start to get a sense of the relationship between genotypes to phenotypes.
- Depending on the expression profiles, we can get information about the cell's function and activity within an organism.
- It is convenient to generate a cell atlas from organisms and group cells according to their gene expression.
- scRNA-seq helps us achieve this.

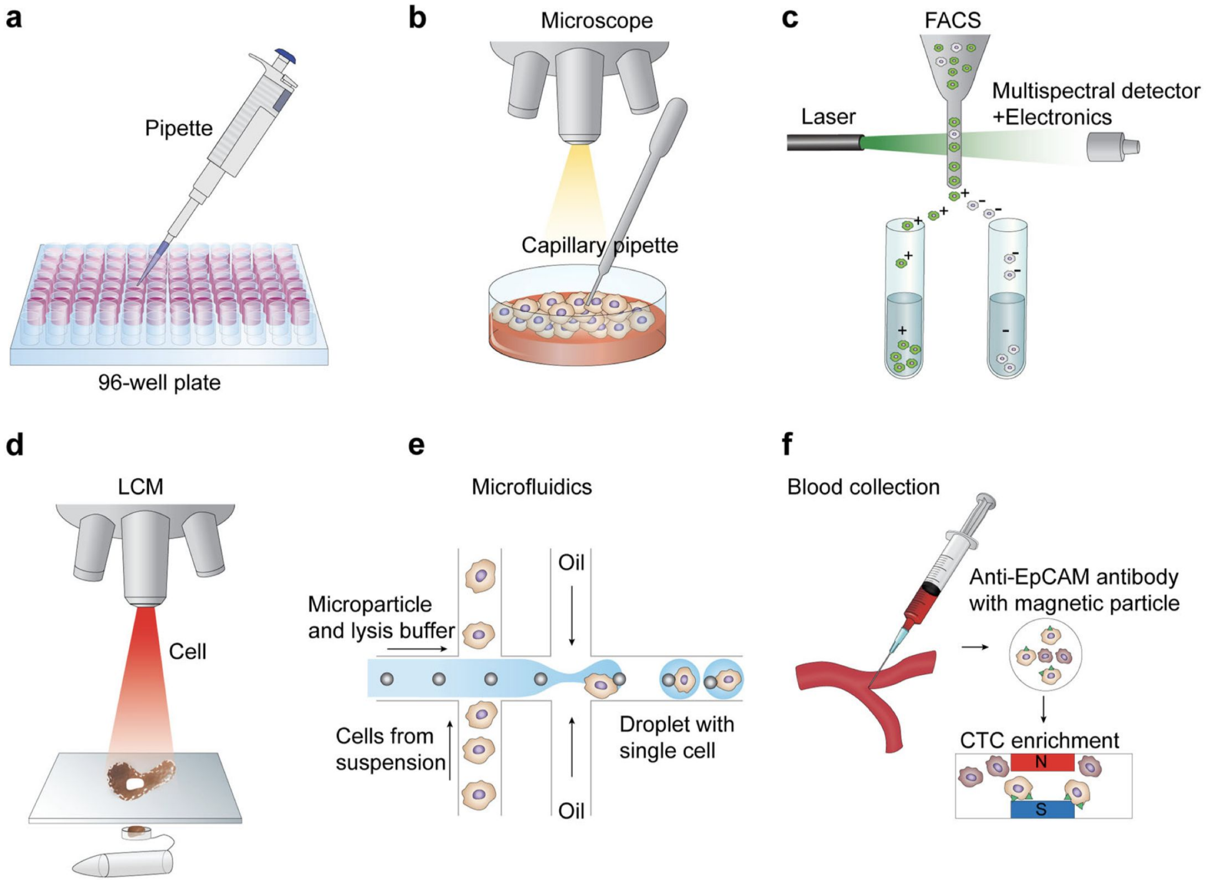
Single-cell RNA-seq



Single Cell RNA Sequencing Workflow

Single cell RNA sequencing workflow. Yijyechern (2014) uploaded to Wikipedia

Single-cell RNA-seq



Single cell isolation techniques

a. Dilution methods
b. Micromanipulation with capillary pipettes
c. FACS
d. Laser capture microdissection
e. Microfluidics
f. Enumeration of circulating tumor cells

Hwang, B., Lee, J. H., & Bang, D. (2018)

Applying machine learning to determine cell types

Unsupervised Learning

- *Cell clustering: Single Cell Consensus Clustering (**SC3**)*

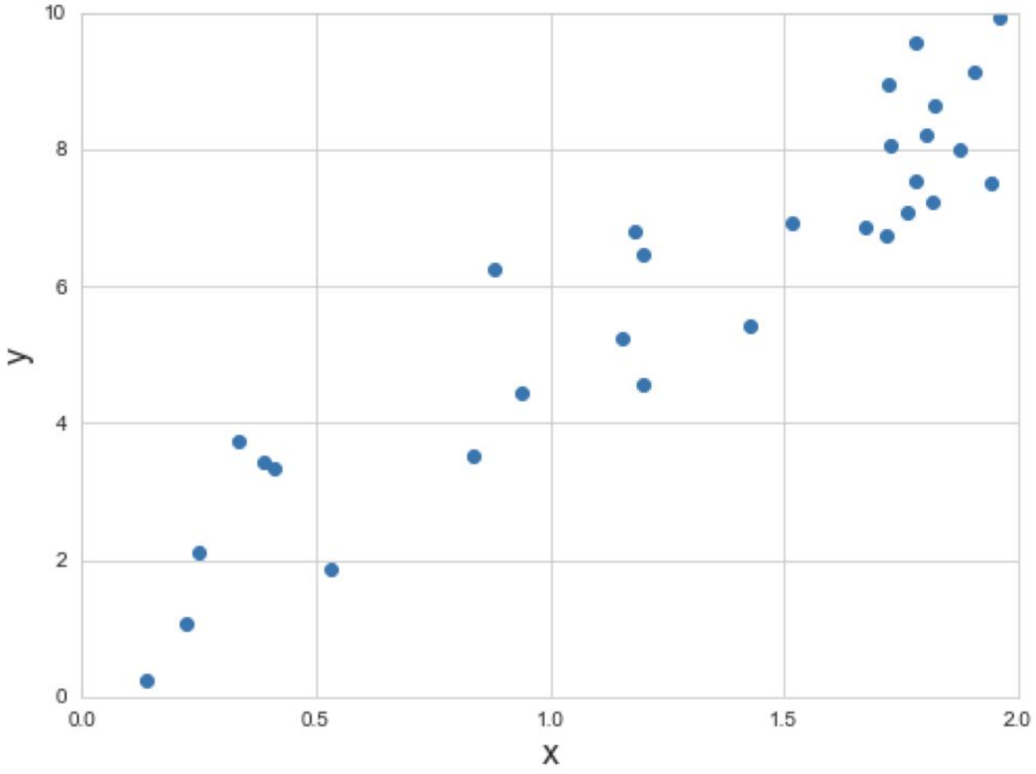
Annotation of cell types

- *Labor Intensive*
- *Revisions needs manual reevaluation of previous annotations*
- *Annotations are made for each dataset and not easily transferred between datasets*

Machine learning background:

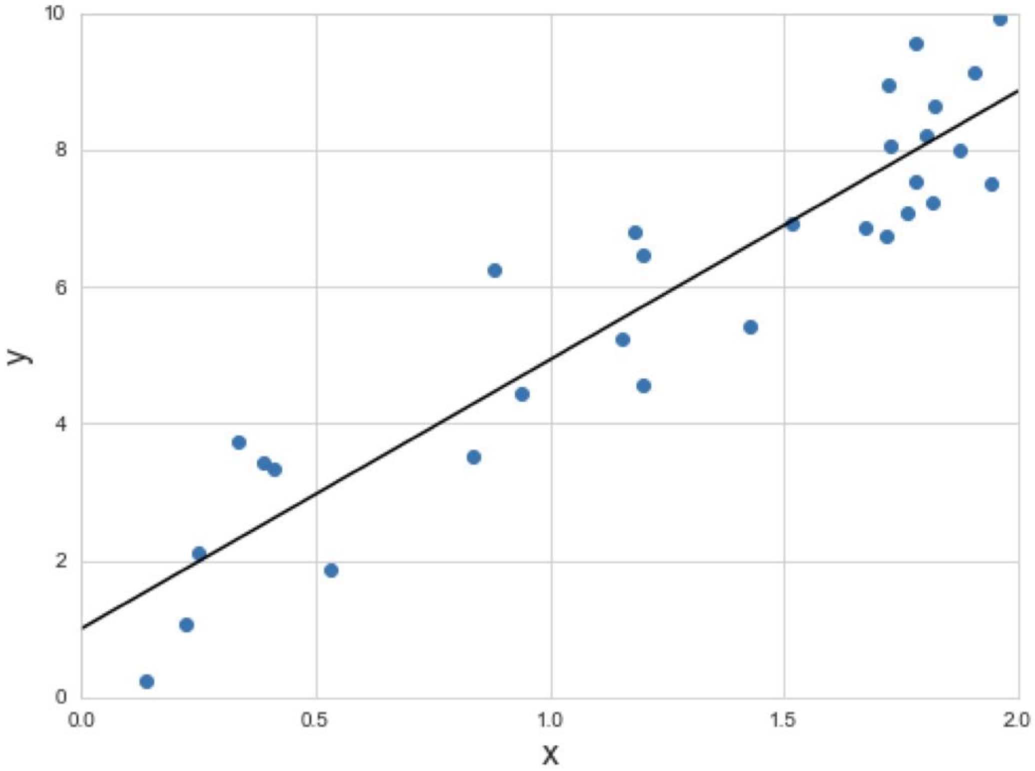
Regression and regularization

Choosing a model



Intuition: there is a linear relationship between some feature *x* and a response *y*

Writing a model

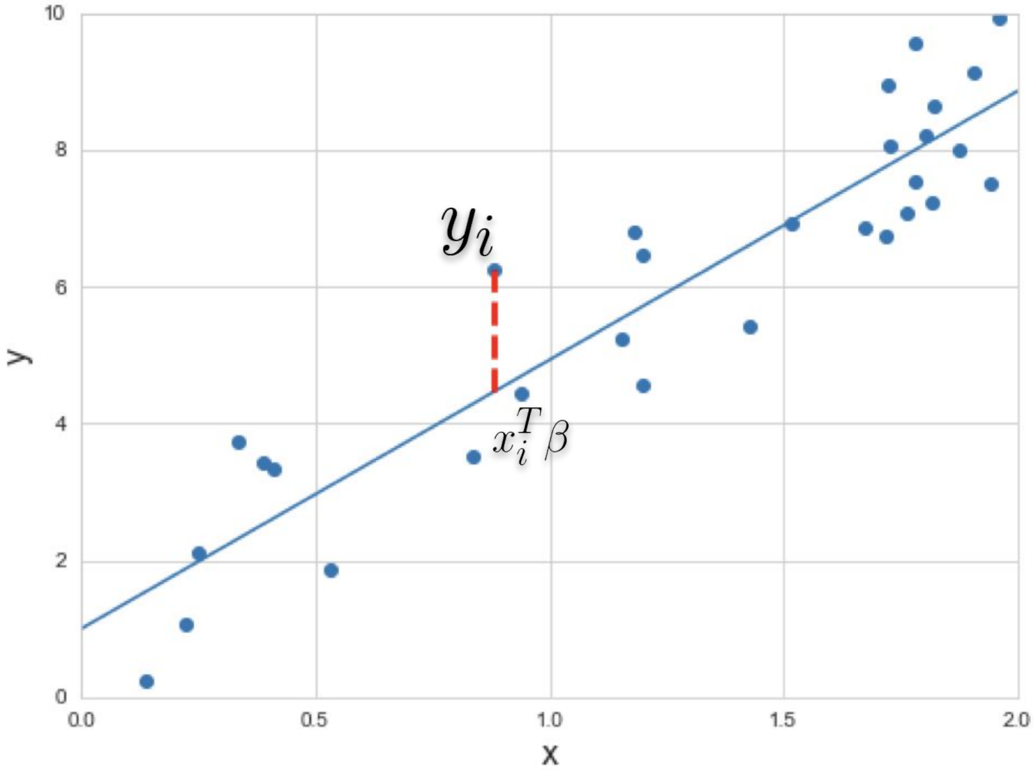


Math: y = mx + b

Vector notation:

$$y = \begin{bmatrix} x & 1 \end{bmatrix} \cdot \begin{bmatrix} m \\ b \end{bmatrix} = x^T \beta$$

Fitting a model



How do we choose the "best" line?

Objective function: sum of squared distances

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T\beta)^2$$

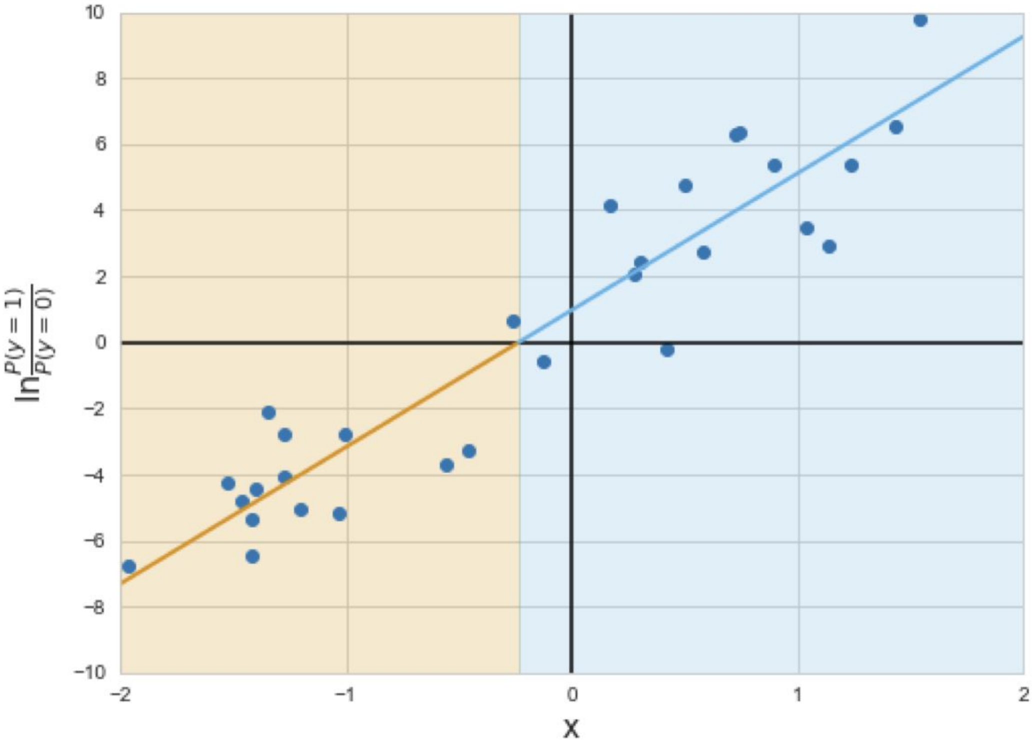
Writing a model - logistic regression

Still a linear relationship with the features!

$$\ln \frac{P(y=1)}{P(y=0)} = x^T\beta$$

But now we're modeling the log-odds ratio of two classes

Logistic regression - interpretation

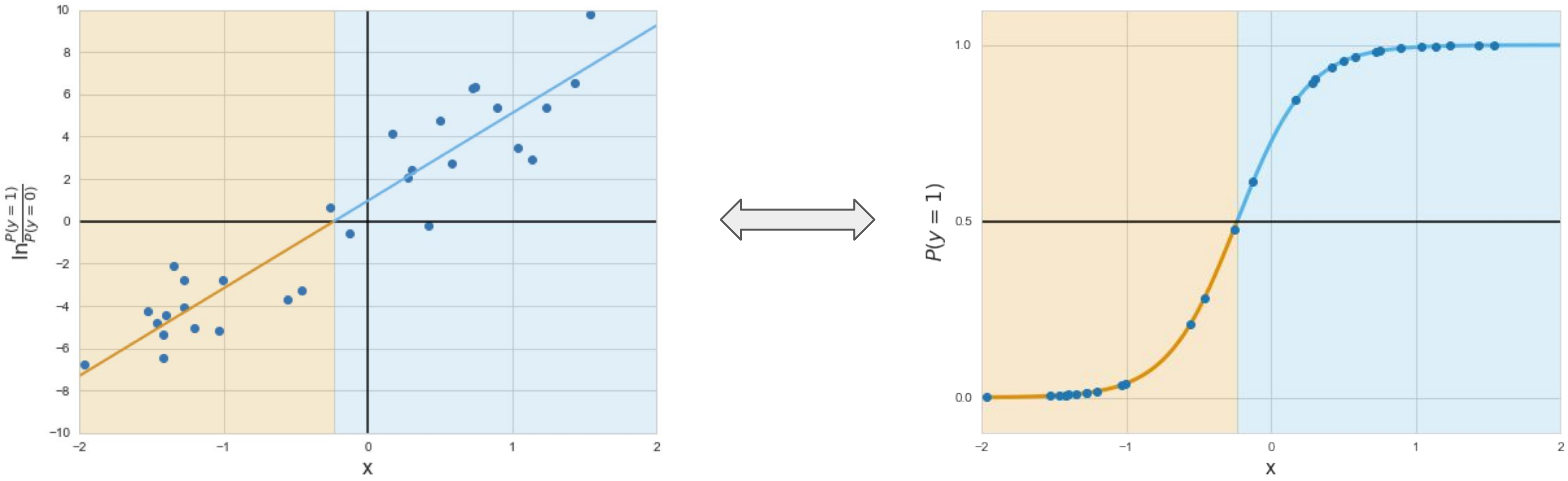


$$\ln \frac{P(y=1)}{P(y=0)} = x^T\beta$$

If class 1 is more likely, this fraction is >1 and the log-odds ratio is positive

If class 0 is more likely, the fraction is <1 and the log-odds ratio is negative

Another view of logistic regression



Some algebra gives us a formula for P(y=1)

Our (x*b>0) decision boundary is equivalent to the point where the probability of class 1 becomes more than 50%

$$P(y=1) = \frac{e^{x^T\beta}}{1+e^{x^T\beta}}$$

The logistic regression cost function

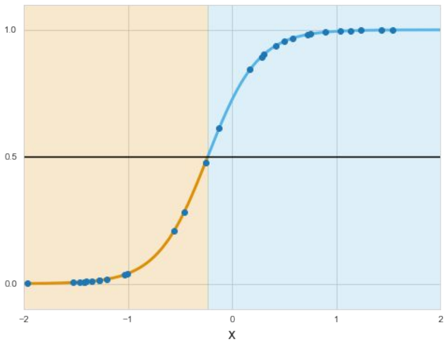
We want to maximize the log-likelihood of our class assignments

$$P(\beta \mid X) = \sum_{i=1}^{n} \{ y_i \log P(y_i = 1) + (1 - y_i) \log P(y_i = 0) \}$$

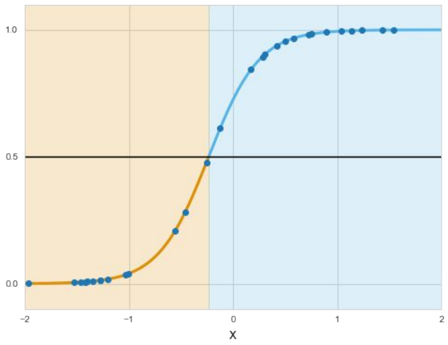
Intuition: sum the log probabilities of the class assignments

Multinomial logistic regression

P(y=1)

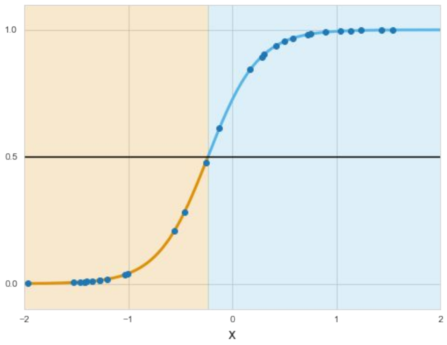


P(y=2)



...

P(y=k)



Intuition:

- for *k* classes, fit *k* binomial logistic models
  - this means we have independent scores for each class
- Choose class assignments that maximize the log-likelihoods

Regularization

Intuition: want to constrain our regression vector (the “slopes” and intercept)

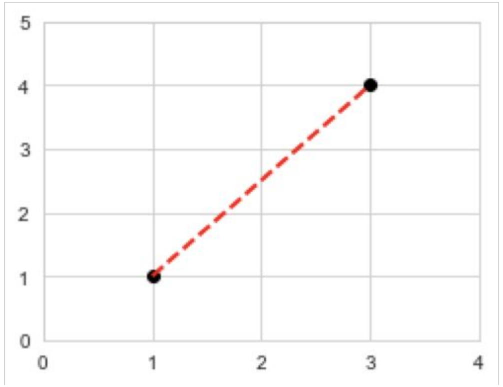
The regularization parameter lets us balance the trade-off between cost function and regularization penalty

$$\hat{\beta} = \arg\min_{\beta} \left(\text{loss} + \lambda \cdot \text{penalty} \right)$$

“Norms” - distance between two points, or size of a vector

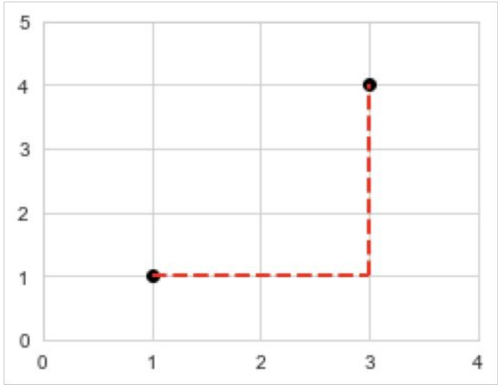
L2-norm
(Euclidean distance)

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$



L1-norm
(Manhattan distance)

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$



Regularized regression

Ridge regression

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \boxed{\lambda ||\beta||_2^2}$$

Shrinks components of the regression vector

Lasso regression

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \boxed{\lambda ||\beta||_1}$$

Sets some components to zero

Elastic net regularization

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \boxed{\left[(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\right]}$$

Mix of ridge and lasso regression

Shrinks groups of correlated components, and removes other groups

New parameter α controls the trade-off between shrinkage and sparsity

Putting it all together - multinomial logistic regression + elastic net

Average likelihood
over all points (cells)

$$\ell(\{\beta_{0k},\beta_k\}_1^K) = -\left[\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K} y_{il}(\beta_{0k}+x_i^T\beta_k) - \log\left(\sum_{k=1}^{K} e^{\beta_{0k}+x_i^T\beta_k}\right)\right)\right] + \lambda\left[(1-\alpha)||\beta||_F^2/2 + \alpha\sum_{j=1}^{p}||\beta_j||_q\right]$$

Likelihood of
belonging to class *k*

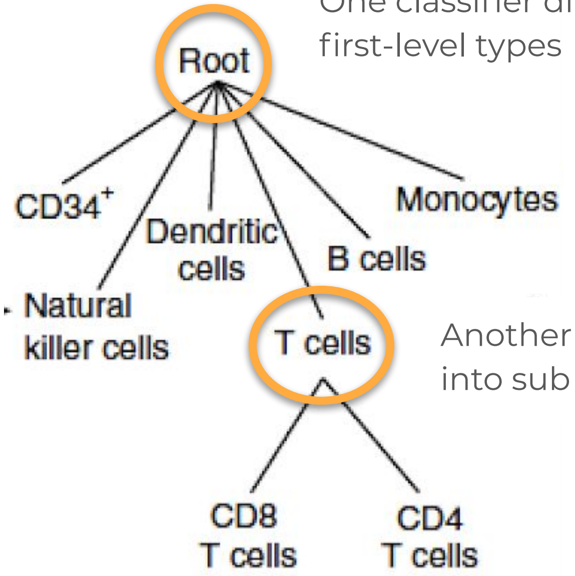
Per-class normalization

Elastic net penalty

GLMnet

Garnett: the algorithm

Big picture



One classifier divides all cells into first-level types

Another classifier divides all T-cells into sub types

Given a cell type hierarchy, fit a multinomial logistic regression model (with elastic net regularization) at each node

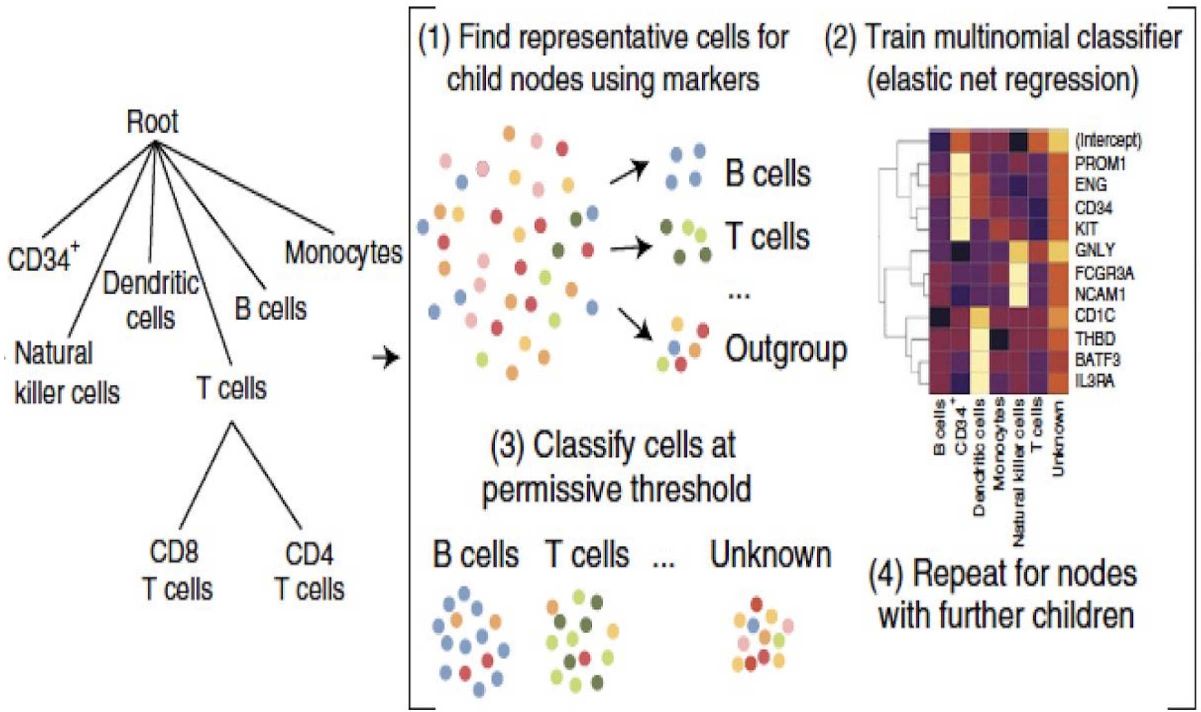
Marker genes define cell type hierarchy

```
>CD34
expressed: CD34, THY1, ENG, KIT, PROM1

>Natural killer cells
expressed: NCAM1, FCGR3A

>B cells
expressed: CD19, MS4A1, CD79A

>T cells
expressed: CD3D, CD3E, CD3G

>CD4 T cells
expressed: CD4, FOXP3, IL2RA, IL7R
subtype of: T cells

>CD8 T cells
expressed: CD8A, CD8B
subtype of: T cells
```

This is the “supervised” part of Garnett

The user specifies a list of marker genes for known cell types

“Representative” cells are used to train each classifier



Each model needs a set of “true” cell type labels

However, we typically don’t know the cell types in our sample (otherwise we wouldn’t need Garnett!)

So, we want to choose groups of cells that look “the most” like known cell types

Choosing representative cells

A cell is “representative” if it scores in the 75th percentile for a cell type and no others

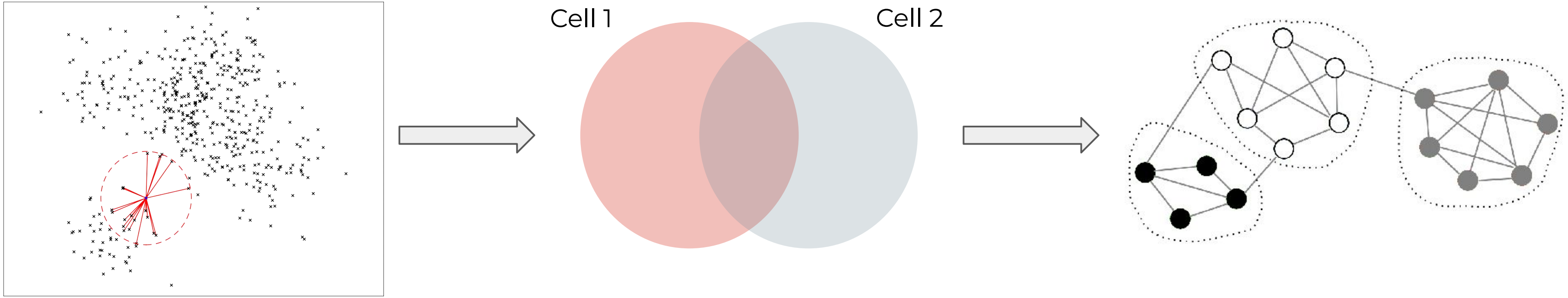
$$S_{c,j} = \sum_{k \in G_c} T_{k,j}$$

- Score how well each cell (*j*) represents each known cell type (*c*)
- $T_{k,j}$ is a “marker score” that represents whether gene *k* was expressed in cell *j*
  - Adjusted per-gene to account for strength and frequency of expression
- Final score is the sum of the adjusted marker scores over all marker genes for a cell type

Clustering cells

Garnett implements a cell clustering algorithm for two purposes

1. Selecting “outgroup” cells for classifier training
2. “Cluster-extended” classification - “majority vote” cell type assignment within clusters



20 nearest neighbors

Jaccard similarity graph

Louvain clustering

How is the model trained?

- Normalized expression for all representative cells (+ outgroup)
- Include genes expressed in >5% of cells in at least one training type
- Exclude genes expressed in 90th percentile of all cell types
- Multinomial regression with elastic net (alpha=0.3)
- NOTE: marker genes always included, and NOT regularized

How are cells classified?

Per-cell (cluster-agnostic):

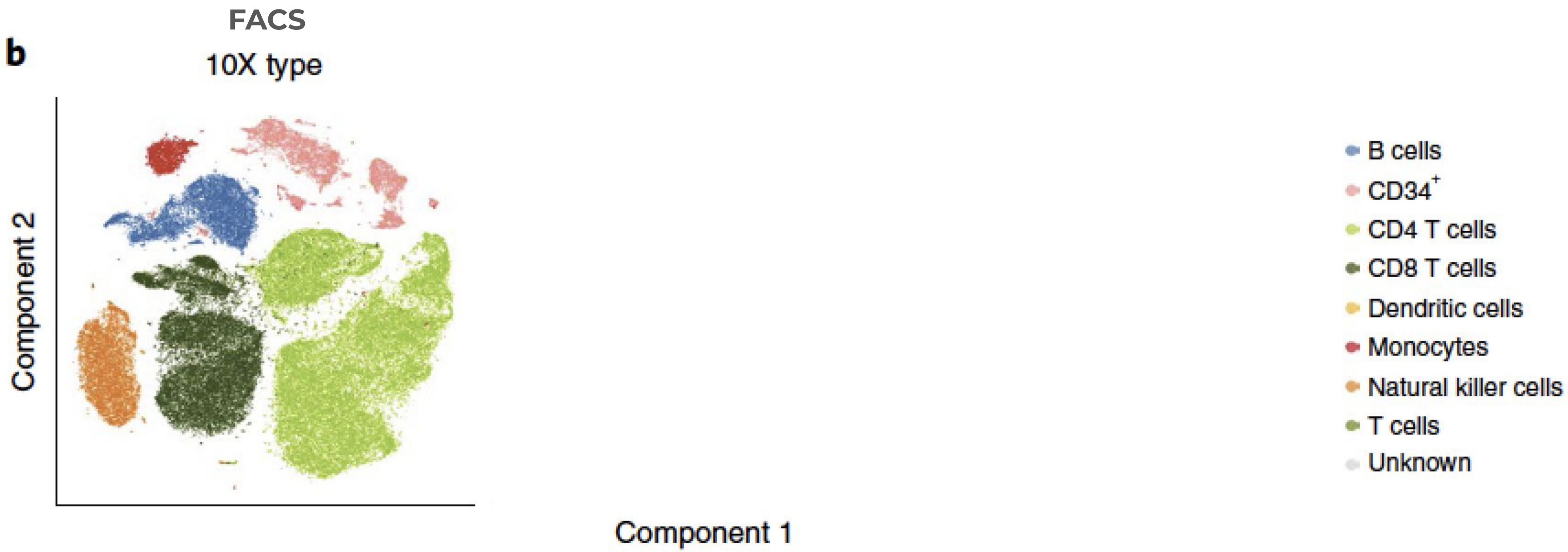
$$\text{type}(j) = c_1 \quad \text{if } \frac{P(c_1)}{P(c_2)} \geq 1.5$$

Cluster-extended:

- Give all cells in cluster same assignment if >5% of cells are classified, and >90% of these have same type

Garnett: the results

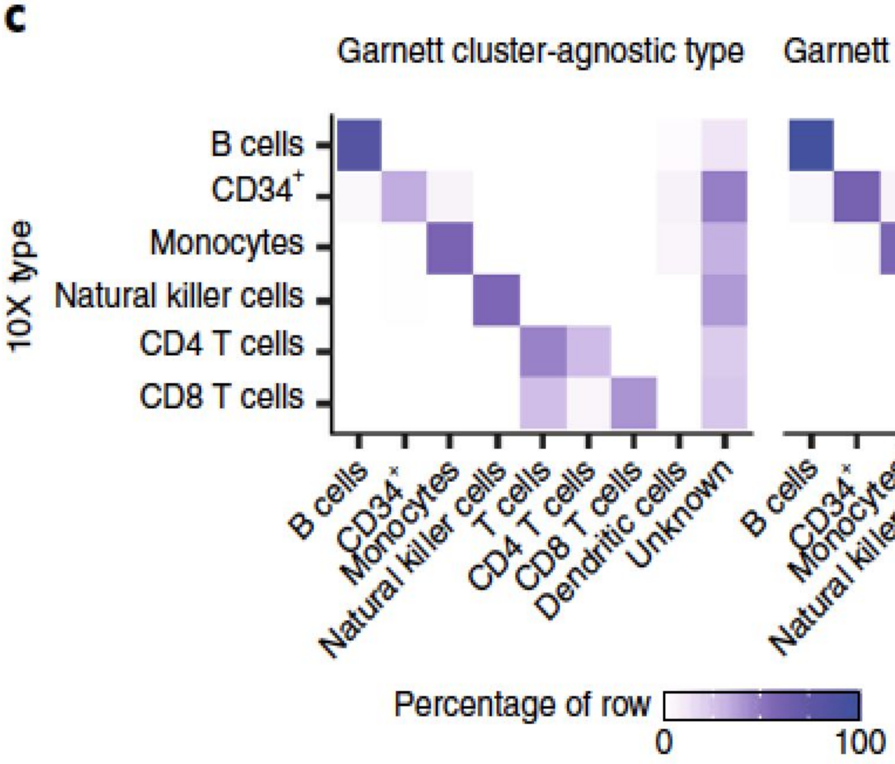
Results: t-SNE plots of PBMC data



Assigned 71% of cells to the correct type (cluster-agnostic); 34% of all T cells received their sub classification.

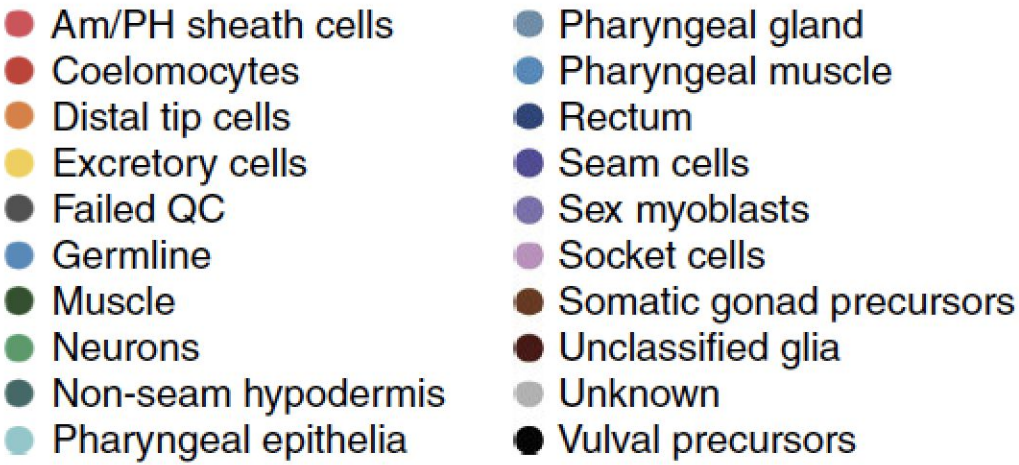
Assigned 94% of cells to the correct type (cluster-extended); 91% of T cells received their subclassification.

Results: Heatmap of PBMC data



Color represents the percentage of cells of a certain FACS type labeled as each type by Garnett.

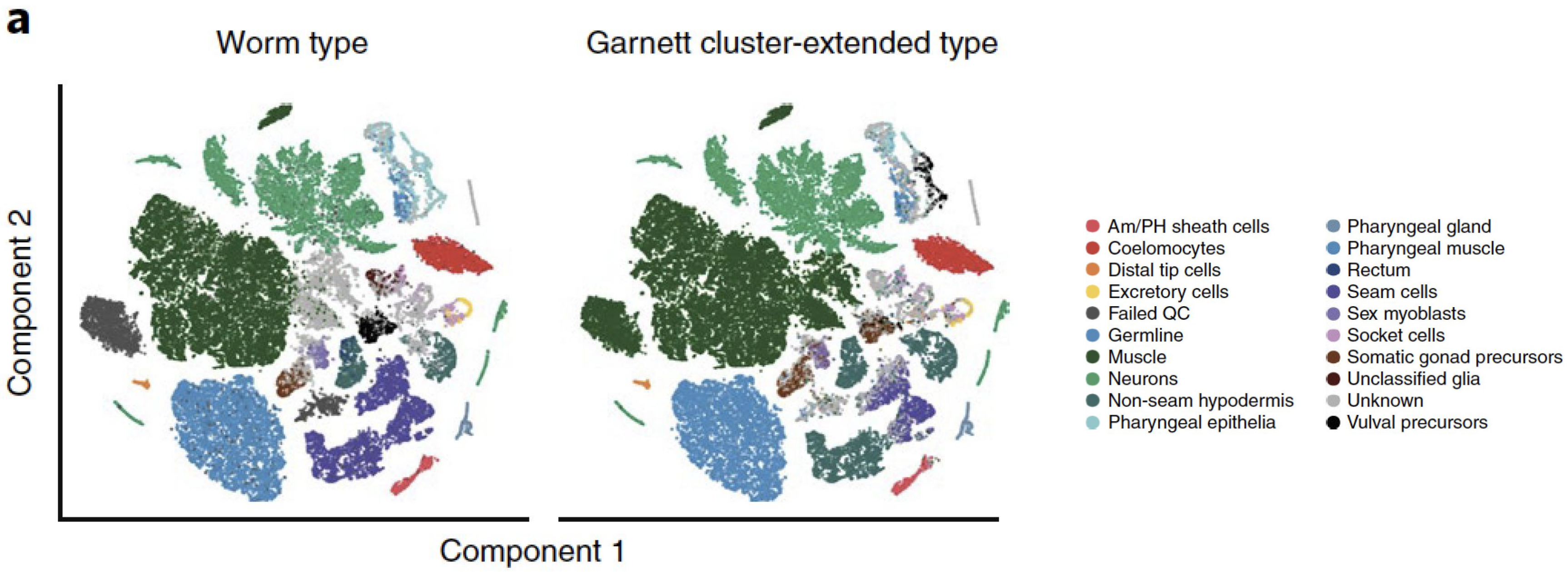
Results: Garnett on *C. elegans* discrimination



- ● Am/PH sheath cells
- ● Coelomocytes
- ● Distal tip cells
- ● Excretory cells
- ● Failed QC
- ● Germline
- ● Muscle
- ● Neurons
- ● Non-seam hypodermis
- ● Pharyngeal epithelia
- ● Pharyngeal gland
- ● Pharyngeal muscle
- ● Rectum
- ● Seam cells
- ● Sex myoblasts
- ● Socket cells
- ● Somatic gonad precursors
- ● Unclassified glia
- ● Unknown
- ● Vulval precursors

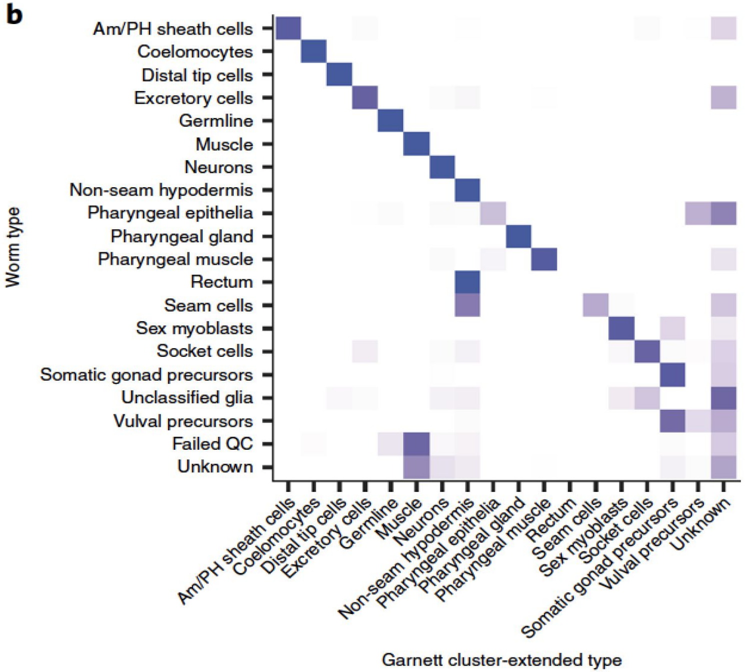
29 major cell types from: **Comprehensive single cell transcriptional profiling of *C. elegans*.** *Published in Science, 2017.*

Results: Garnett on *C. elegans* discrimination, t-SNE

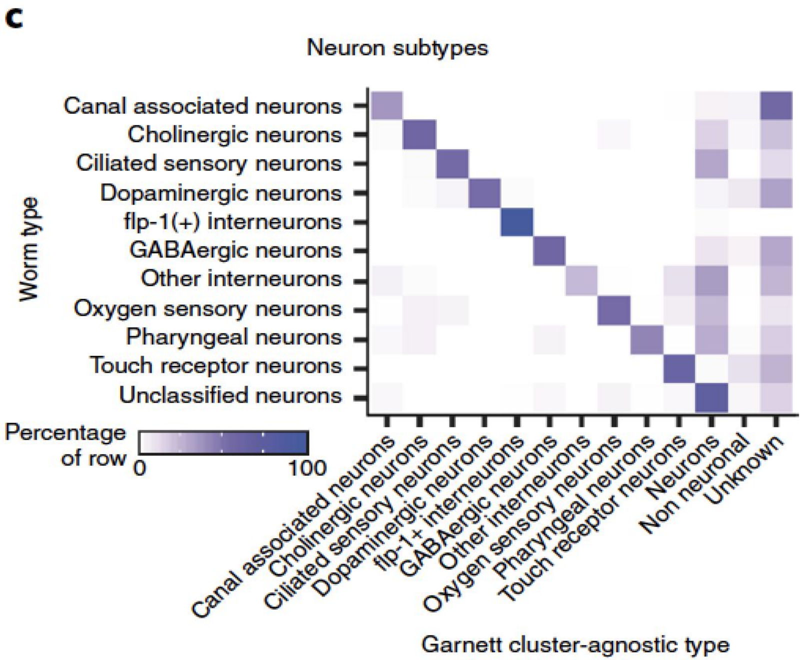


Assigned about 87% of cells correctly

Results: Garnett on *C. elegans* discrimination, Heatmap

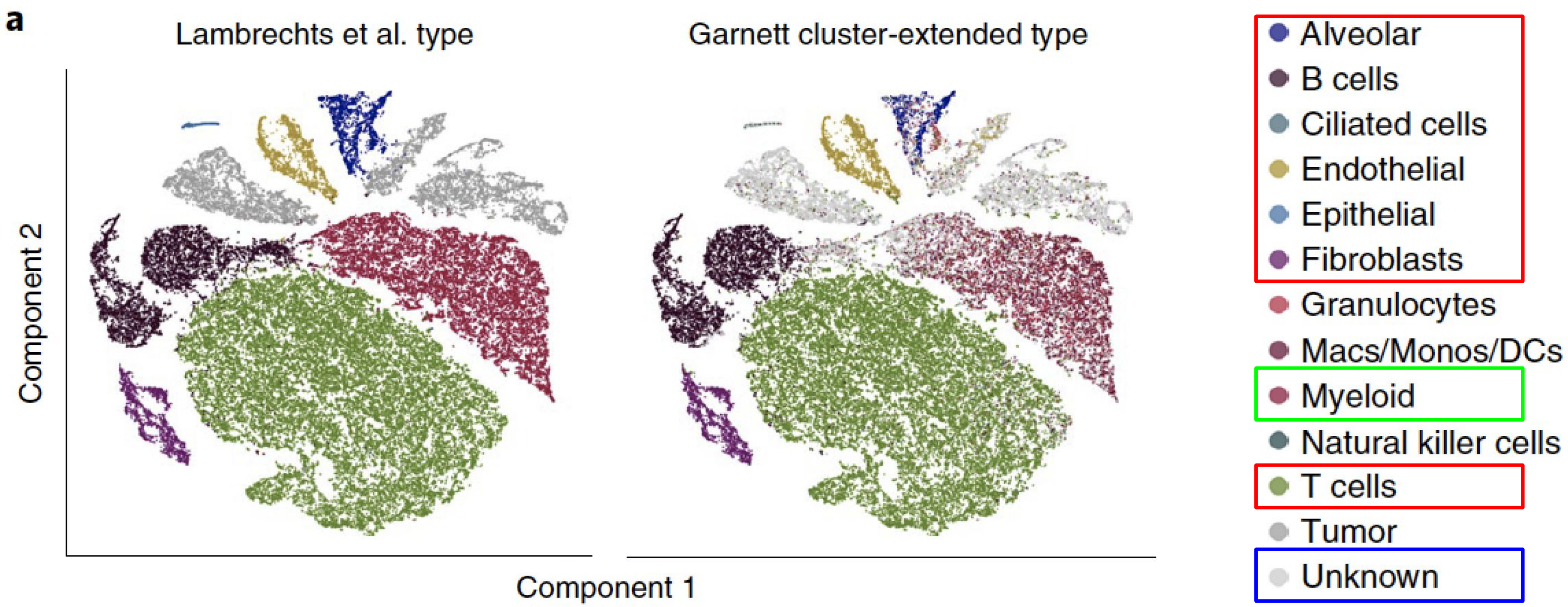


b. Major cell types (~87%)



c. Subtypes (neurons) ~53%

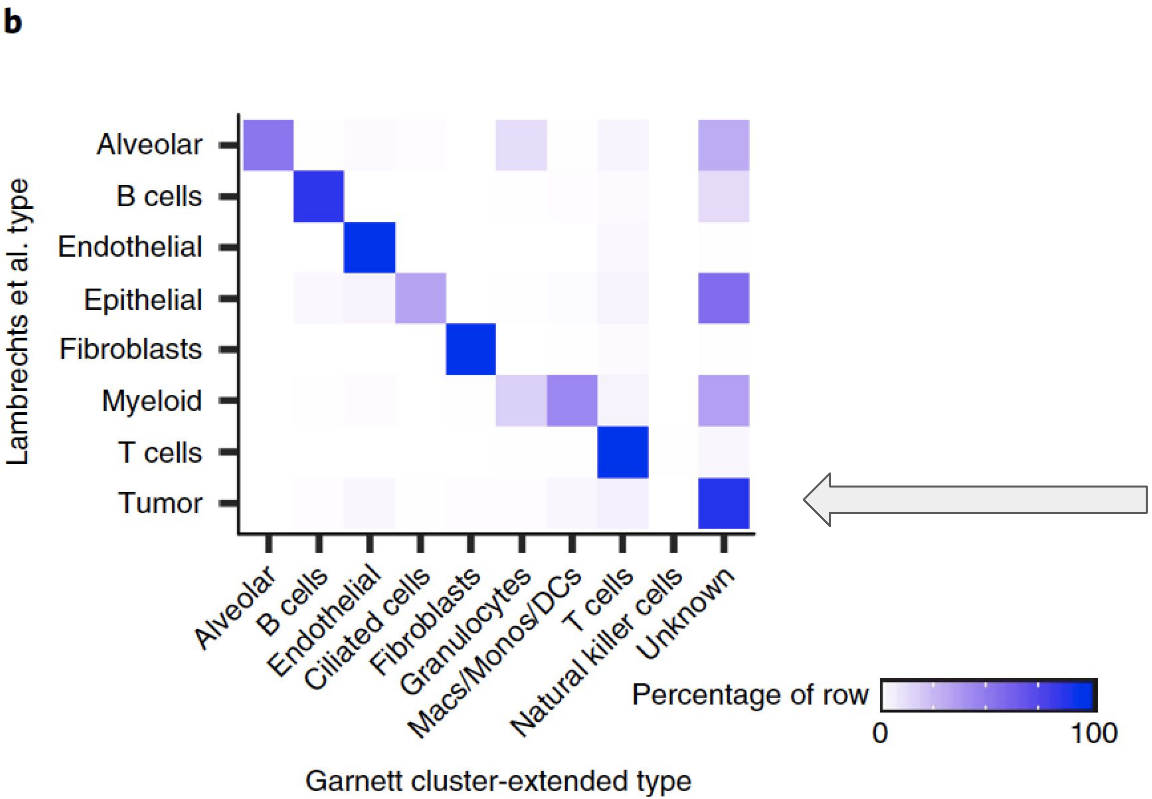
Results: Mouse-trained classifiers on human datasets



Red box: ~92% correctly assigned.   Blue box: ~55% were tumor cells

Green Box: Garnett labeled ~44% as Macs/Monos/DS from MCA myeloid

Results: Mouse-trained classifiers on human datasets



Unknown cells in Garnett were actually reported as tumon cells

Conclusions

Novel single-cell sequencing technologies require the ability to delineate cell types

Garnett trains a multinomial logistic regression model on cells that are “representative” of user provided cell types

Garnett accurately classifies cell types from a variety of species

Open questions

How does the hierarchical classification work? I.e., how does Garnett determine whether a cell should be assigned a sub type or its parent?

A: if ratio test for sub type results in “unknown”, keep parent assignment

How is the outgroup used during classification? Cells are annotated unknown if the ratio between first and second choice assignments is too small; what if first choice is the outgroup?

A: not clear, but probably also “unknown”