Exponential-family embedding with application to cell developmental trajectories for single-cell RNA-seq data

Kevin Lin

with Kathryn Roeder and Jing Lei

We are interested in studying the developmental trajectory of oligodendrocytes using single-cell RNA-seq data.

Previous technology:
Many cells investigated at once

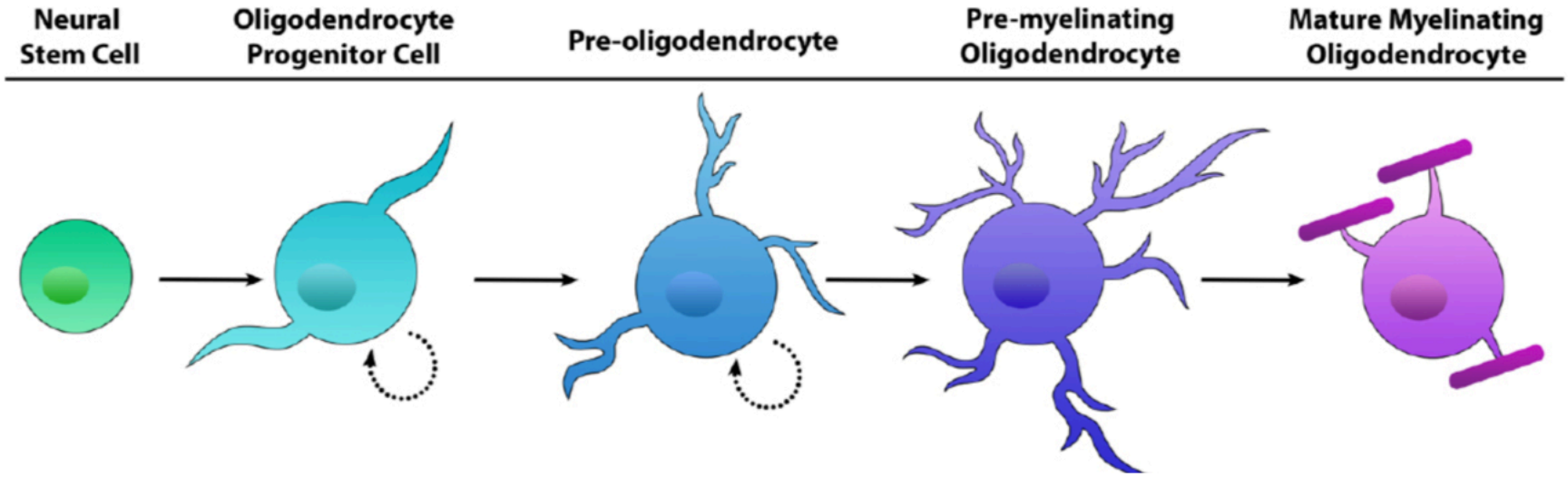
New technology (Popularized 2015+):
Single cell

We are interested in studying the developmental trajectory of oligodendrocytes using single-cell RNA-seq data.

- Oligodendrocytes: rapid transmission of signals and provide metabolic support to neurons in the central nervous system

We are interested in studying the developmental trajectory of oligodendrocytes using single-cell RNA-seq data.

- Oligodendrocytes: rapid transmission of signals and provide metabolic support to neurons in the central nervous system



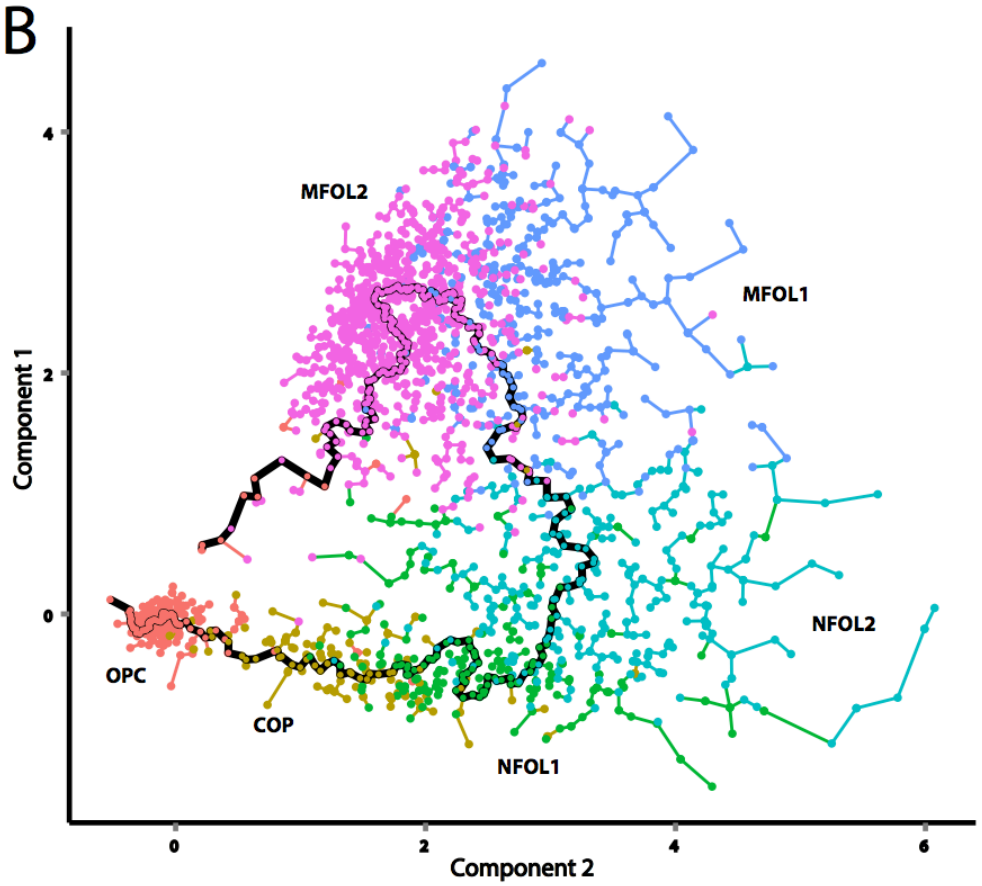
We use the dataset from Marques et al. (2016), which divides the cells into 6 major cell types and 13 sub-types.

Major Cell Type (6 total)	Count (5000+ total)	Number of cell sub-types (13 total)
Pdgfra+ precursor		
Oligodendrocyte precursor cell		
Differentiation-committed oligodendrocyte precursors		
Newly formed oligodendrocytes		
Myelin-forming oligodendrocytes		
Mature oligodendrocyte		

We use the dataset from Marques et al. (2016), which divides the cells into 6 major cell types and 13 sub-types.

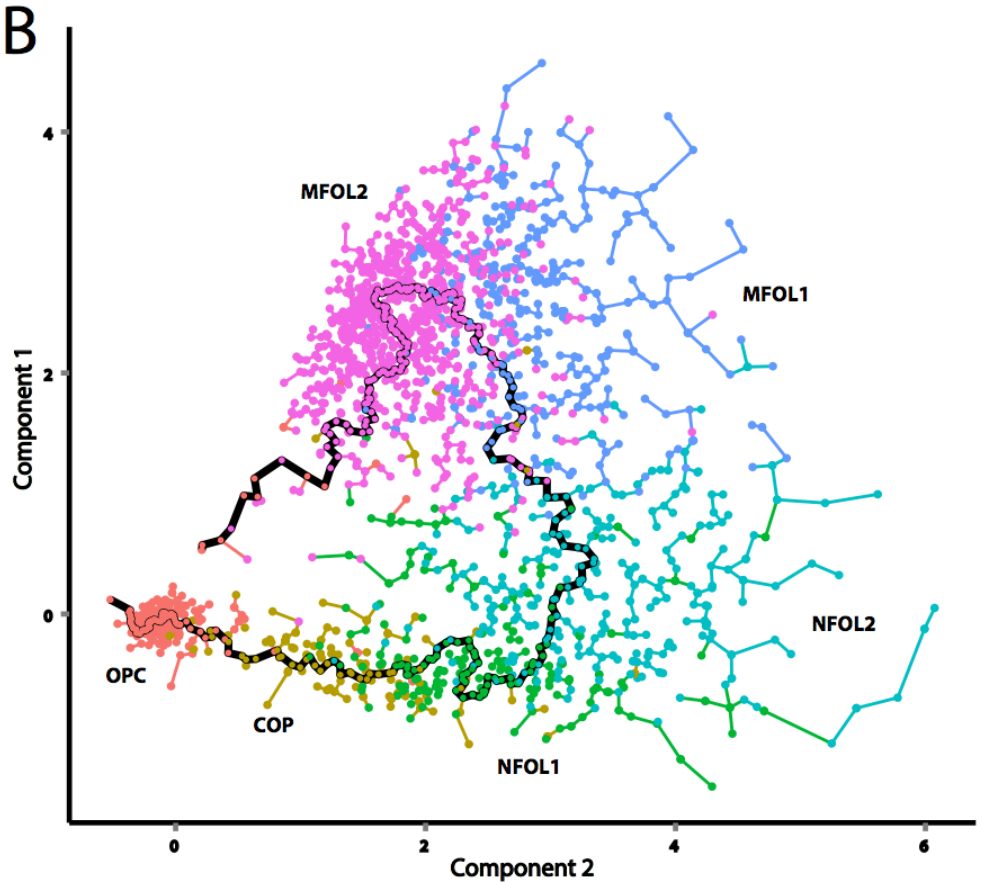
Major Cell Type (6 total)	Count (5000+ total)	Number of cell sub-types (13 total)
Pdgfra+ precursor	70+	1
Oligodendrocyte precursor cell	300+	1
Differentiation-committed oligodendrocyte precursors	100+	1
Newly formed oligodendrocytes	500+	2
Myelin-forming oligodendrocytes	1200+	2
Mature oligodendrocyte	2700+	6

We use the dataset from Marques et al. (2016), which divides the cells into 6 major cell types and 13 sub-types.



Past work (Marques et al. 2016)

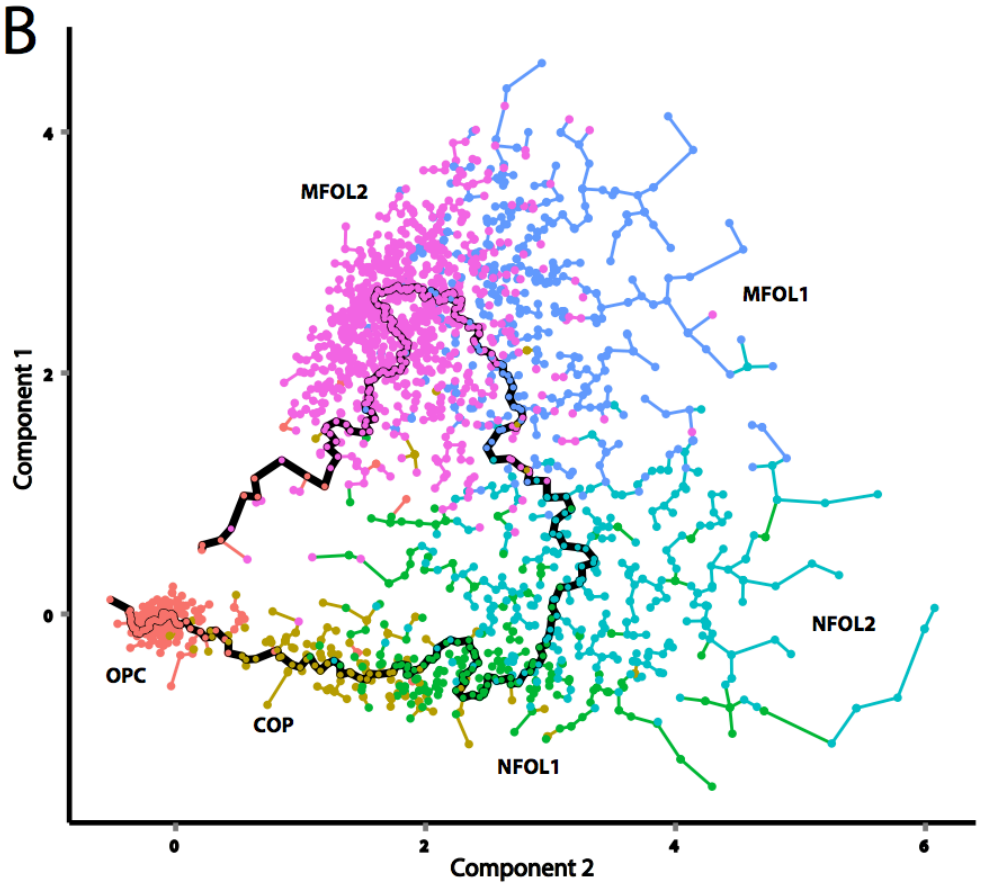
We use the dataset from Marques et al. (2016), which divides the cells into 6 major cell types and 13 sub-types.



Past work (Marques et al. 2016)

- Two major components:
  - Embedding each cell into a lower dimensional space
  - Estimating the developmental trajectories

We use the dataset from Marques et al. (2016), which divides the cells into 6 major cell types and 13 sub-types.



Past work (Marques et al. 2016)

- Two major components:
  - Embedding each cell into a lower dimensional space
  - Estimating the developmental trajectories

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$A_{ij} \sim F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)$$

n cells and *p* genes

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$\underbrace{A_{ij}}_{\text{Single-cell data}} \sim F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)$$

n cells and *p* genes

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$A_{ij} \sim \underbrace{F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)}$$

Exponential-family distribution

$$p(A_{ij} \mid \theta_{ij}) \propto \exp\left(A_{ij}^\top \theta_{ij} - g(\theta_{ij})\right)$$

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$A_{ij} \sim \underbrace{F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)}$$

Exponential-family distribution

$$p(A_{ij} \mid \underbrace{\theta_{ij}}) \propto \exp\left(A_{ij}^\top \theta_{ij} - g(\theta_{ij})\right)$$

Natural parameter

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

Latent low-dimensional distr. of interest

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$A_{ij} \sim F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)$$

Exponential-family distribution

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

Latent low-dimensional distr. of interest

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$A_{ij} \sim F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)$$

Exponential-family distribution

ZIFA (Pierson et al. 2015): Gaussian
ZINB-WaVE (Risso et al. 2018): Negative binomial
pCMF (Durif et al. 2017): Poisson

To estimate this low-dimensional embedding, we impose a hierarchical, latent space model.

Latent low-dimensional distr. of interest

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \overset{i.i.d.}{\sim} G, \quad \text{and} \quad \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p \overset{i.i.d.}{\sim} H$$

$$A_{ij} \sim F\left(\theta_{ij} = \boldsymbol{X}_i^\top \boldsymbol{Y}_j\right)$$

Exponential-family distribution

Natural idea: For a given distribution family *F*, find the factorization that maximizes the log-likelihood based on ***A*** in order to recover *G*

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

Minimize over two low-rank matrices:

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{np} \sum_{(i,j)} \left[g(\boldsymbol{X}_i^\top \boldsymbol{Y}_j) - A_{ij}^\top (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right]$$

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

Minimize over two low-rank matrices:

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{np} \sum_{(i,j)} \left[g(\boldsymbol{X}_i^\top \boldsymbol{Y}_j) - A_{ij}^\top (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right]$$

Plugging in the appropriate form for fixed-variance Gaussian

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

Minimize over two low-rank matrices:

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{np} \sum_{(i,j)} \left[g(\boldsymbol{X}_i^\top \boldsymbol{Y}_j) - A_{ij}^\top (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right]$$

Solved exactly by SVD

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

- Diagnostic based on matrix completion (omit 2% of values)

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

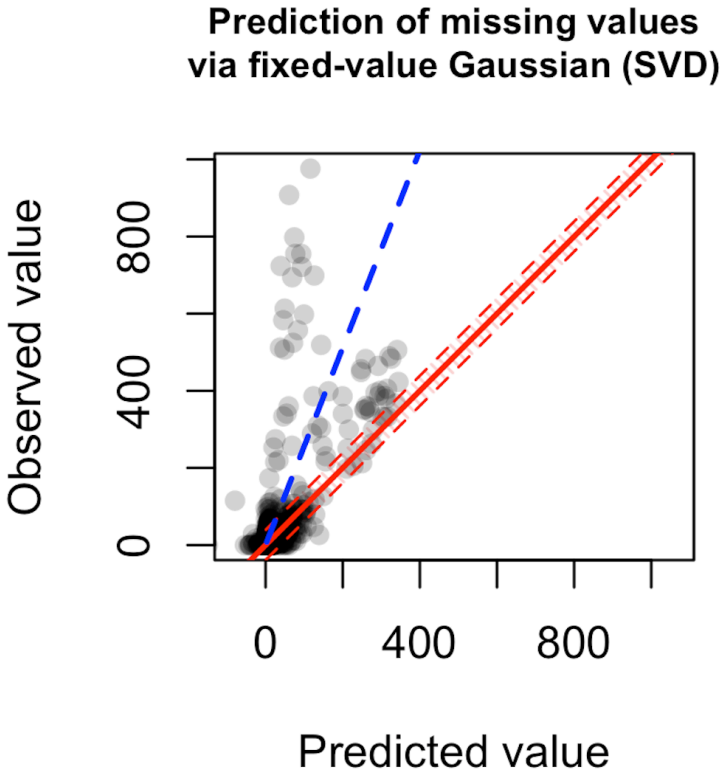
SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

- Diagnostic based on matrix completion (omit 2% of values)
- Using SoftImpute, the matrix completion counterpart to SVD

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

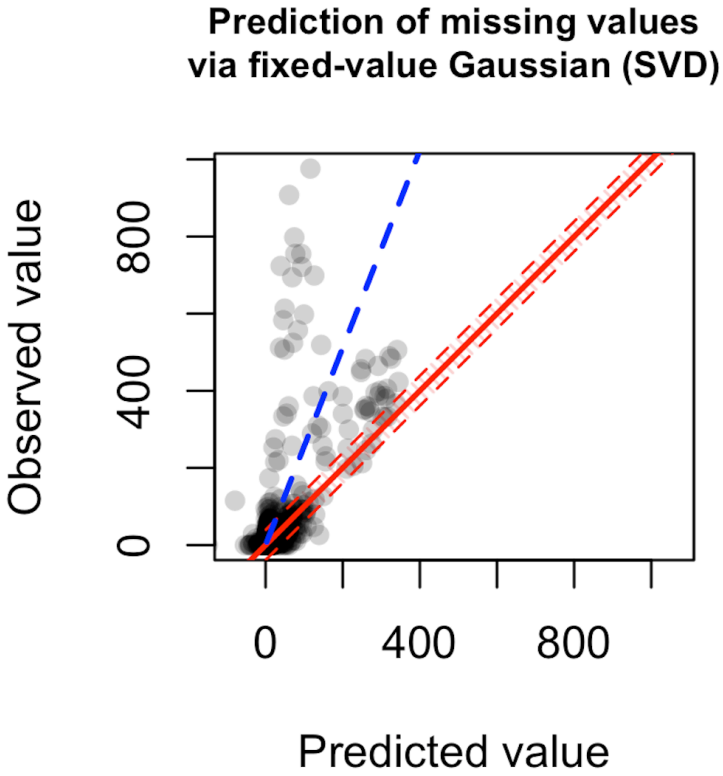
- Diagnostic based on matrix completion (omit 2% of values)
- Using SoftImpute, the matrix completion counterpart to SVD



$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

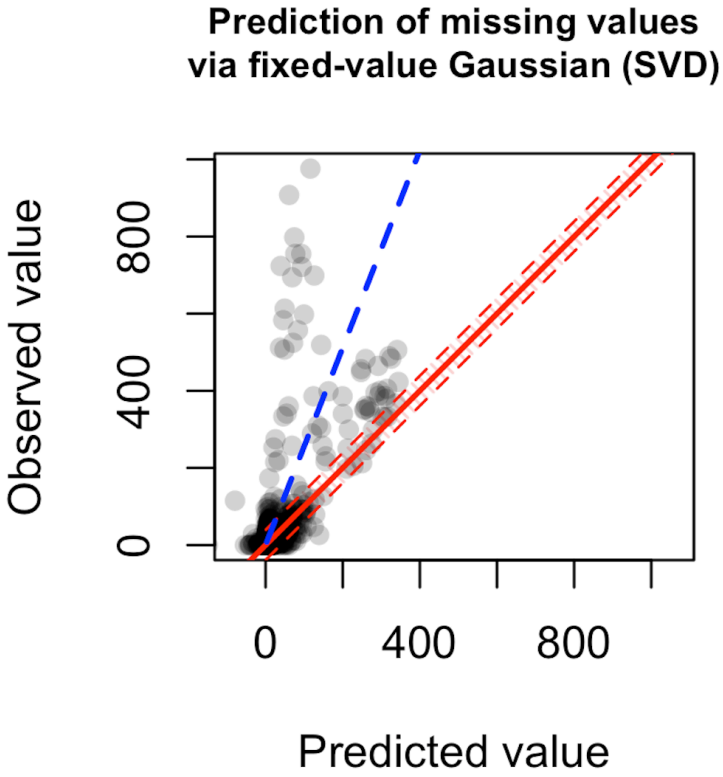
- Blue: Principal component vector between predicted and observed values



$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

SVD's implicitly assumes a Gaussian model with fixed variance, which does not seem to be a great model.

- Blue: Principal component vector between predicted and observed values
- Red: Theoretical band if model were correctly specified



$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) \propto \frac{1}{np} \sum_{(i,j)} \left[\left(A_{ij} - (\boldsymbol{X}_i^\top \boldsymbol{Y}_j) \right)^2 \right]$$

We provide a more flexible way to embed cells into a lower-dimensional space, backed by statistical theory.

- **Statistical goal:**
  - Estimate the latent vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ (and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p$) from the observed matrix $\boldsymbol{A}$ where each entry A_{ij} is drawn from an exponential-family distribution parameterized by the inner product $\boldsymbol{X}_i^\top \boldsymbol{Y}_j$

We provide a more flexible way to embed cells into a lower-dimensional space, backed by statistical theory.

- **Statistical goal:**
  - Estimate the latent vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ (and $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_p$) from the observed matrix $\boldsymbol{A}$ where each entry A_{ij} is drawn from an exponential-family distribution parameterized by the inner product $\boldsymbol{X}_i^\top \boldsymbol{Y}_j$
- **Scientific goal:**
  - Apply our method to aid with estimating oligodendrocytes' lineages

Method

(eSVD, for exponential-family SVD)

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 1: Initialization
  - Initialize the estimate based on SVD of some transformation of A

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 1: Initialization
  - Initialize the estimate based on SVD of some transformation of A
- Stage 2: Alternating minimization
  - Update the estimate by re-estimating either X, keeping Y fixed, or re-estimating Y, keeping X fixed

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 1: Initialization
  - Initialize the estimate based on SVD of some transformation of A
- Stage 2: Alternating minimization
  - Update the estimate by re-estimating either X, keeping Y fixed, or re-estimating Y, keeping X fixed
- Stage 3: Reparameterization
  - Ensure identifiability conditions are met

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\boldsymbol{X}^{(t)} = \underset{\boldsymbol{X} \in \mathbb{R}^{n \times k}}{\arg\min}\, \mathcal{L}(\boldsymbol{X}, \overline{\boldsymbol{Y}}^{(t)}) \;:\; \boldsymbol{X}_i^\top \overline{\boldsymbol{Y}}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

$$\boldsymbol{Y}^{(t+1)} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times k}}{\arg\min}\, \mathcal{L}(\overline{\boldsymbol{X}}^{(t+1)}, \boldsymbol{Y}) \;:\; (\overline{\boldsymbol{X}}_i^{(t+1)})^\top \boldsymbol{Y}_j \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{Y}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{Y}^{(t+1)})$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\boldsymbol{X}^{(t)} = \underset{\boldsymbol{X} \in \mathbb{R}^{n \times k}}{\arg\min}\, \mathcal{L}(\boldsymbol{X}, \overline{\boldsymbol{Y}}^{(t)}) \quad : \quad \boldsymbol{X}_i^\top \overline{\boldsymbol{Y}}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

$$\boldsymbol{Y}^{(t+1)} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times k}}{\arg\min}\, \mathcal{L}(\overline{\boldsymbol{X}}^{(t+1)}, \boldsymbol{Y}) \quad : \quad (\overline{\boldsymbol{X}}_i^{(t+1)})^\top \boldsymbol{Y}_j \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{Y}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{Y}^{(t+1)})$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\boldsymbol{X}^{(t)} = \underset{\boldsymbol{X} \in \mathbb{R}^{n \times k}}{\arg\min}\, \mathcal{L}(\boldsymbol{X}, \overline{\boldsymbol{Y}}^{(t)})$$

Minimizing one of the factors based on the log-likelihood (Convex problem)

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

$$\boldsymbol{Y}^{(t+1)} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times k}}{\arg\min}\, \mathcal{L}(\overline{\boldsymbol{X}}^{(t+1)}, \boldsymbol{Y})$$

$$\overline{\boldsymbol{Y}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{Y}^{(t+1)})$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\boldsymbol{X}^{(t)} = \underset{\boldsymbol{X} \in \mathbb{R}^{n \times k}}{\arg\min}\, \mathcal{L}(\boldsymbol{X}, \overline{\boldsymbol{Y}}^{(t)}) \;:\; \boldsymbol{X}_i^\top \overline{\boldsymbol{Y}}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

$$\boldsymbol{Y}^{(t+1)} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times k}}{\arg\min}\, \mathcal{L}(\overline{\boldsymbol{X}}^{(t+1)}, \boldsymbol{Y}) \;:\; (\overline{\boldsymbol{X}}_i^{(t+1)})^\top \boldsymbol{Y}_j \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{Y}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{Y}^{(t+1)})$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$X^{(t)} =$$
$$\overline{X}^{(t+1)} =$$
$$Y^{(t+1)} =$$
$$\overline{Y}^{(t+1)} =$$

Need to ensure the inner product still yields a valid natural parameter

We use Frank-Wolfe to do this constrained optimization

$$\boldsymbol{X}_i^\top \overline{\boldsymbol{Y}}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$: (\overline{\boldsymbol{X}}_i^{(t+1)})^\top \boldsymbol{Y}_j \in \mathcal{R}, \quad \forall (i,j),$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$X^{(t)} = \arg\min_{X \in \mathbb{R}^{n \times k}} \mathcal{L}(X, \overline{Y}^{(t)}) \ : \ X_i^\top \overline{Y}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{X}^{(t+1)} = \text{LeftSVD}(X^{(t)}),$$

$$Y^{(t+1)} = \arg\min_{Y \in \mathbb{R}^{p \times k}} \mathcal{L}(\overline{X}^{(t+1)}, Y) \ : \ (\overline{X}_i^{(t+1)})^\top Y_j \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{Y}^{(t+1)} = \text{LeftSVD}(Y^{(t+1)})$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

We reparameterize each estimate by its left singular vectors for identifiability reasons

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\boldsymbol{X}^{(t)} = \underset{\boldsymbol{X} \in \mathbb{R}^{n \times k}}{\arg\min}\, \mathcal{L}(\boldsymbol{X}, \overline{\boldsymbol{Y}}^{(t)}) \;:\; \boldsymbol{X}_i^\top \overline{\boldsymbol{Y}}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

$$\boldsymbol{Y}^{(t+1)} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times k}}{\arg\min}\, \mathcal{L}(\overline{\boldsymbol{X}}^{(t+1)}, \boldsymbol{Y}) \;:\; (\overline{\boldsymbol{X}}_i^{(t+1)})^\top \boldsymbol{Y}_j \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{Y}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{Y}^{(t+1)})$$

We use alternating minimization for this non-convex problem, inspired by the matrix factorization literature.

- Stage 2: Alternating minimization

$$\boldsymbol{X}^{(t)} = \underset{\boldsymbol{X} \in \mathbb{R}^{n \times k}}{\arg\min}\, \mathcal{L}(\boldsymbol{X}, \overline{\boldsymbol{Y}}^{(t)}) \;:\; \boldsymbol{X}_i^\top \overline{\boldsymbol{Y}}_j^{(t)} \in \mathcal{R}, \quad \forall (i,j),$$

$$\overline{\boldsymbol{X}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{X}^{(t)}),$$

$$\boldsymbol{Y}^{(t+1)} = \underset{\boldsymbol{Y} \in \mathbb{R}^{p \times k}}{\arg\min}\, \mathcal{L}(\overline{\boldsymbol{X}}^{(t+1)}, \boldsymbol{Y})$$

$$\overline{\boldsymbol{Y}}^{(t+1)} = \text{LeftSVD}(\boldsymbol{Y}^{(t+1)})$$

We then move to the next factor. We keep alternating between the two factors until convergence.

Brief interlude: We use techniques from various papers to show the convergence of the embedding.

Brief interlude: We use techniques from various papers to show the convergence of the embedding.

- Estimating matrix of natural parameters:
  - Requires understanding what properties the initialization needs to satisfy as well as smoothness/convexity of the objective function (Balakrishnan et al. 2017, Zhao et al. 2015)

Brief interlude: We use techniques from various papers to show the convergence of the embedding.

- Estimating matrix of natural parameters:
  - Requires understanding what properties the initialization needs to satisfy as well as smoothness/convexity of the objective function (Balakrishnan et al. 2017, Zhao et al. 2015)

- Handling the extra source of randomness due to the hierarchical model:
  - Requires careful study of identifiability conditions and perturbation bounds (Lei 2019)

Brief interlude: We use techniques from various papers to show the convergence of the embedding.

- Estimating matrix of natural parameters:
  - Requires understanding what properties the initialization needs to satisfy as well as smoothness/convexity of the objective function (Balakrishnan et al. 2017, Zhao et al. 2015)
- Handling the extra source of randomness due to the hierarchical model:
  - Requires careful study of identifiability conditions and perturbation bounds (Lei 2019)
- Applying these generic bounds to a specific model:
  - Requires deploying statistical machinery from other high-dimensional papers (Sur et al. 2019)

Results

(We estimate two lineages for oligodendrocytes.)

We used the curved Gaussian distribution to model our data.

$$A_{ij} \sim F = \text{Normal}(\mu,\ \mu^2/4)$$

We used the curved Gaussian distribution to model our data.

$$A_{ij} \sim F = \text{Normal}(\mu,\ \mu^2/4)$$

so the natural parameter becomes:

$$\boldsymbol{X}_i^\top \boldsymbol{Y}_j = \theta_{ij} = -1/\mu$$

We used the curved Gaussian distribution to model our data.

$$A_{ij} \sim F = \text{Normal}(\mu,\ \mu^2/4)$$

so the natural parameter becomes:

$$\boldsymbol{X}_i^\top \boldsymbol{Y}_j = \theta_{ij} = -1/\mu$$

- Most of the distribution's mass is positive
- The constant 4 can be tuned
- Spreads out the mass near 0

We used the curved Gaussian distribution to model our data.

$$A_{ij} \sim F = \text{Normal}(\mu,\ \mu^2/4)$$

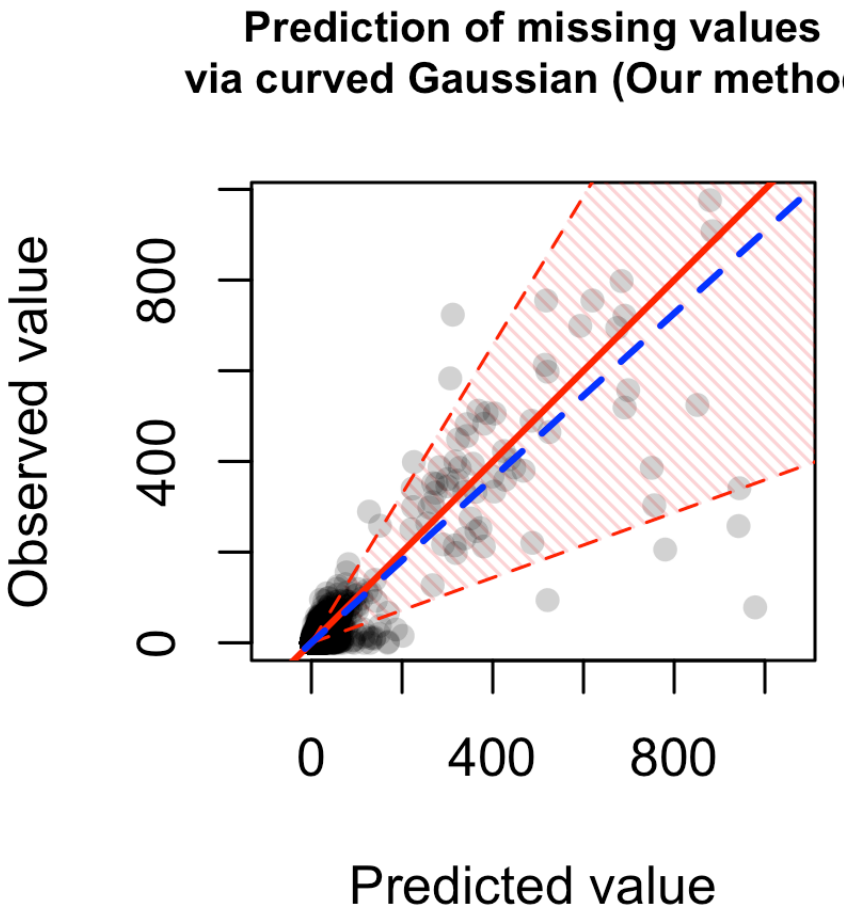
so the natural parameter becomes:

$$\boldsymbol{X}_i^\top \boldsymbol{Y}_j = \theta_{ij} = -1/\mu$$

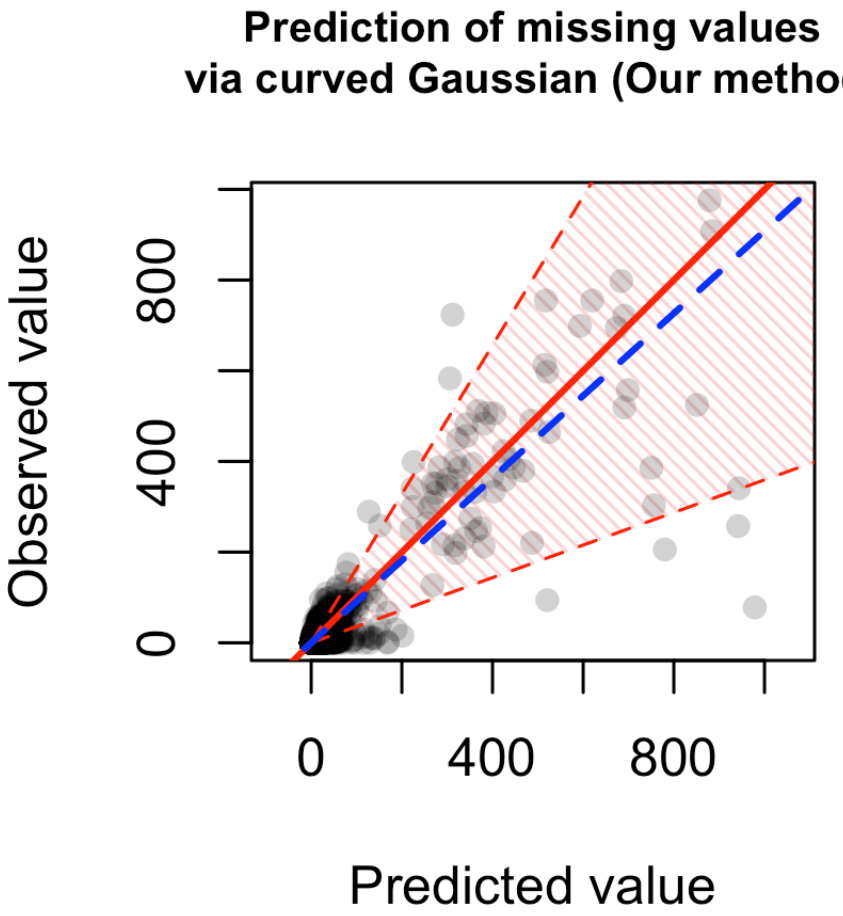
- Most of the distribution's mass is positive
- The constant 4 can be tuned
- Spreads out the mass near 0

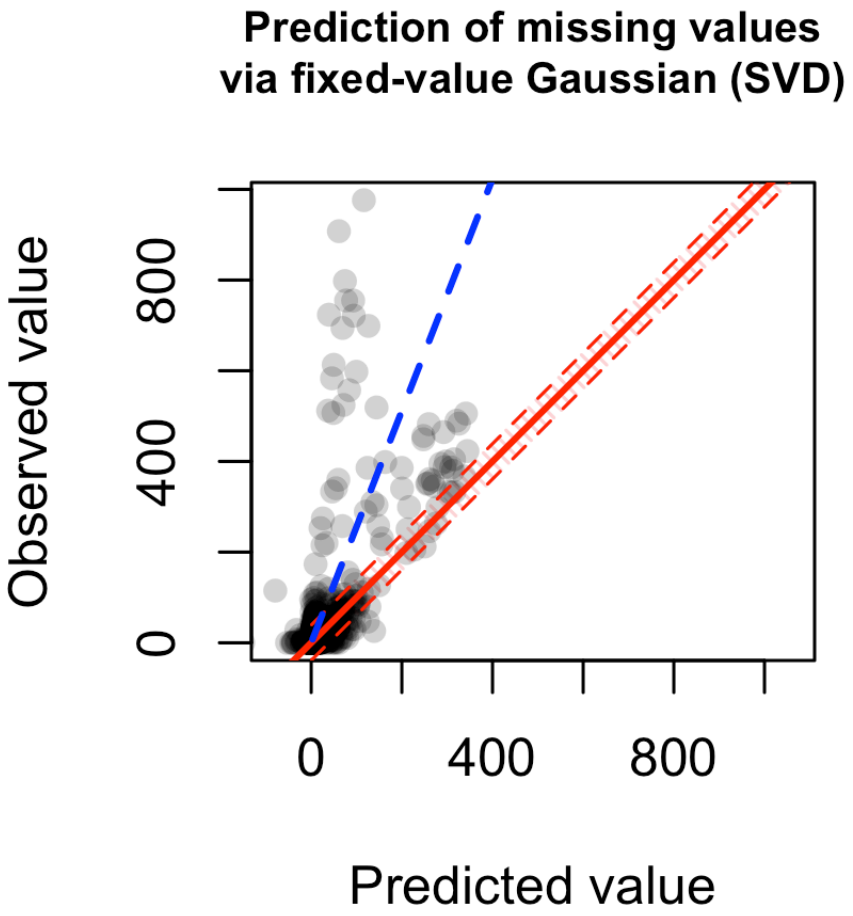
- Recall: To use eSVD, you simply need to plug in the appropriate terms from the exponential family distribution that model *F*.

We used the curved Gaussian distribution to model our data.

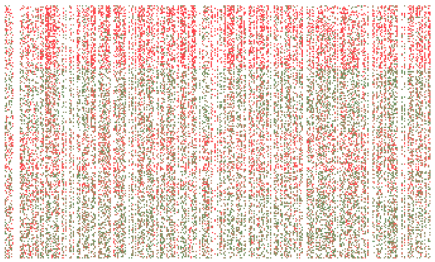


We used the curved Gaussian distribution to model our data.



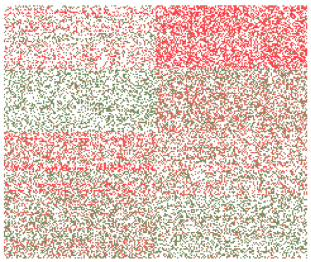


In our pipeline, we screen to keep only the important genes, depth-normalize but do not impute the data.



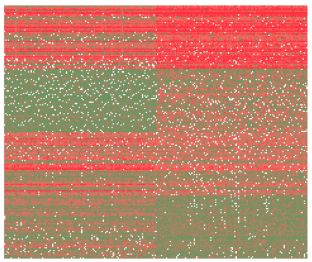
Single-cell RNA-seq data

(Cells as rows, genes as columns)



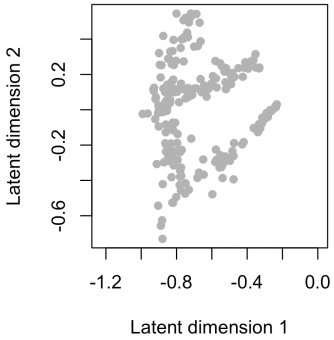
1. Screening relevant genes

(Sparse PCA (Whitten et al., 2009) and DESCEND (Zhu et al., 2018))

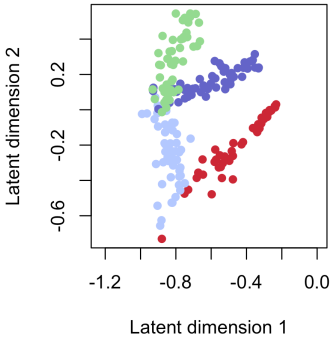


2. Imputing suspected dropout values

(scImpute (Li and Li, 2018))

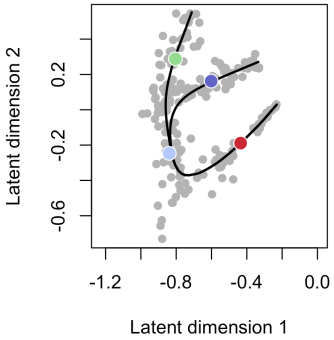


3. Embedding each cell into a lower-dimensional space



4. Clustering the cells

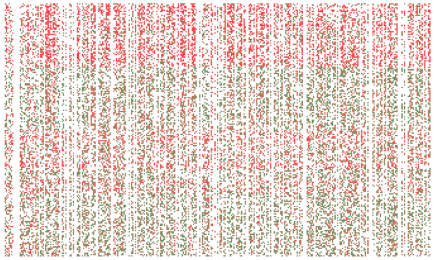
(DBSCAN (Ester et al., 1996))



5. Estimating the trajectories

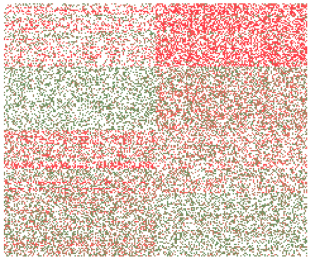
(Slingshot (Street et al., 2018))

In our pipeline, we screen to keep only the important genes, depth-normalize but do not impute the data.



Single-cell RNA-seq data

(Cells as rows, genes as columns)

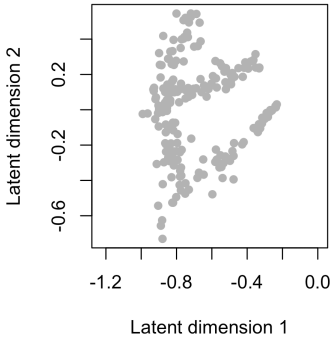


1. Screening relevant genes

(Sparse PCA (Whitten et al., 2009) and DESCEND (Zhu et al., 2018))

2. Imputing suspected dropout values

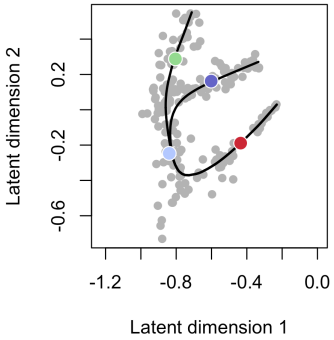
(scImpute (Li and Li, 2018))



3. Embedding each cell into a lower-dimensional space

4. Clustering the cells

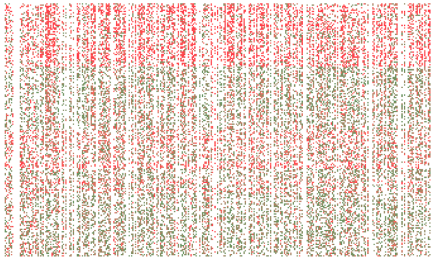
(DBSCAN (Ester et al., 1996))



5. Estimating the trajectories

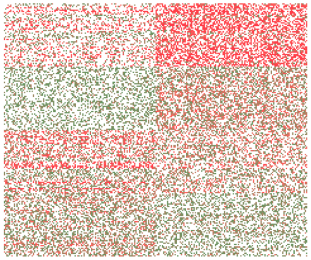
(Slingshot (Street et al., 2018))

In our pipeline, we screen to keep only the important genes, depth-normalize but do not impute the data.



Single-cell RNA-seq data

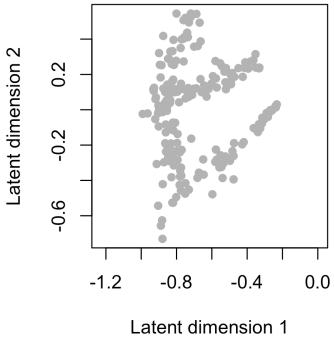
(Cells as rows, genes as columns)



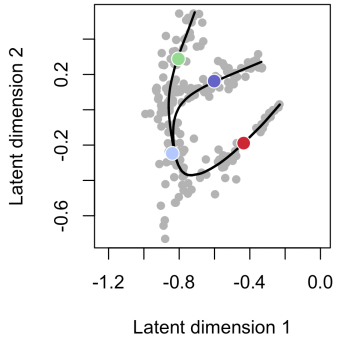
1. Screening relevant genes

(Sparse PCA (Whitten et al., 2009) and DESCEND (Zhu et al., 2018))

We ignore imputation, inspired by Svensson (2019).



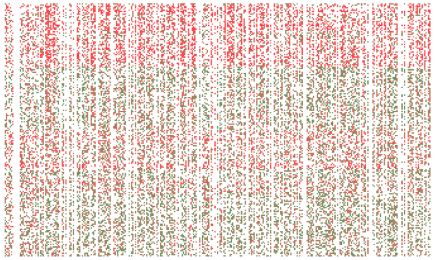
3. Embedding each cell into a lower-dimensional space



5. Estimating the trajectories

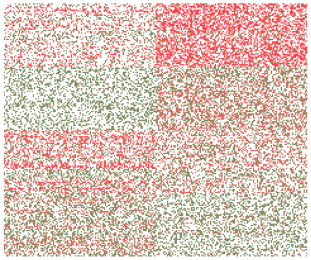
(Slingshot (Street et al., 2018))

In our pipeline, we screen to keep only the important genes, depth-normalize but do not impute the data.



Single-cell RNA-seq data

(Cells as rows, genes as columns)

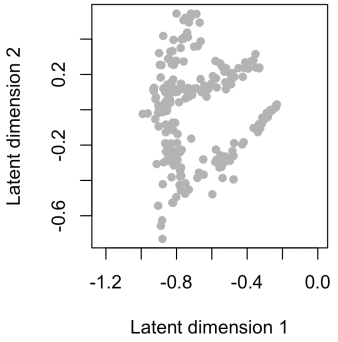


1. Screening relevant genes

(Sparse PCA (Whitten et al., 2009) and DESCEND (Zhu et al., 2018))

We ignore imputation, inspired by Svensson (2019).

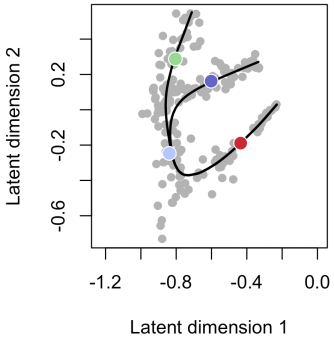
Our work



3. Embedding each cell into a lower-dimensional space

4. Clustering the cells

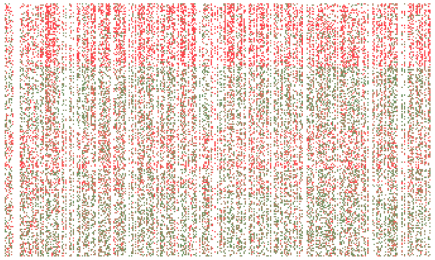
(DBSCAN (Ester et al., 1996))



5. Estimating the trajectories

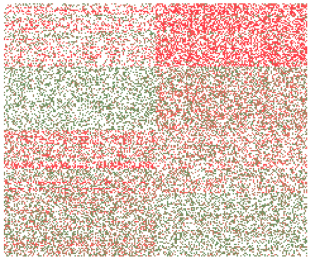
(Slingshot (Street et al., 2018))

In our pipeline, we screen to keep only the important genes, depth-normalize but do not impute the data.



Single-cell RNA-seq data

(Cells as rows, genes as columns)

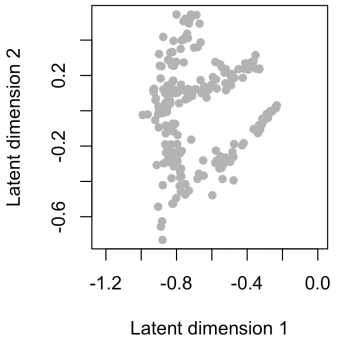


1. Screening relevant genes

(Sparse PCA (Whitten et al., 2009) and DESCEND (Zhu et al., 2018))

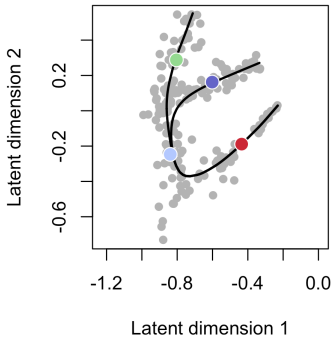
We ignore imputation, inspired by Svensson (2019).

Our work



3. Embedding each cell into a lower-dimensional space

We use the cell sub-types as the given clustering



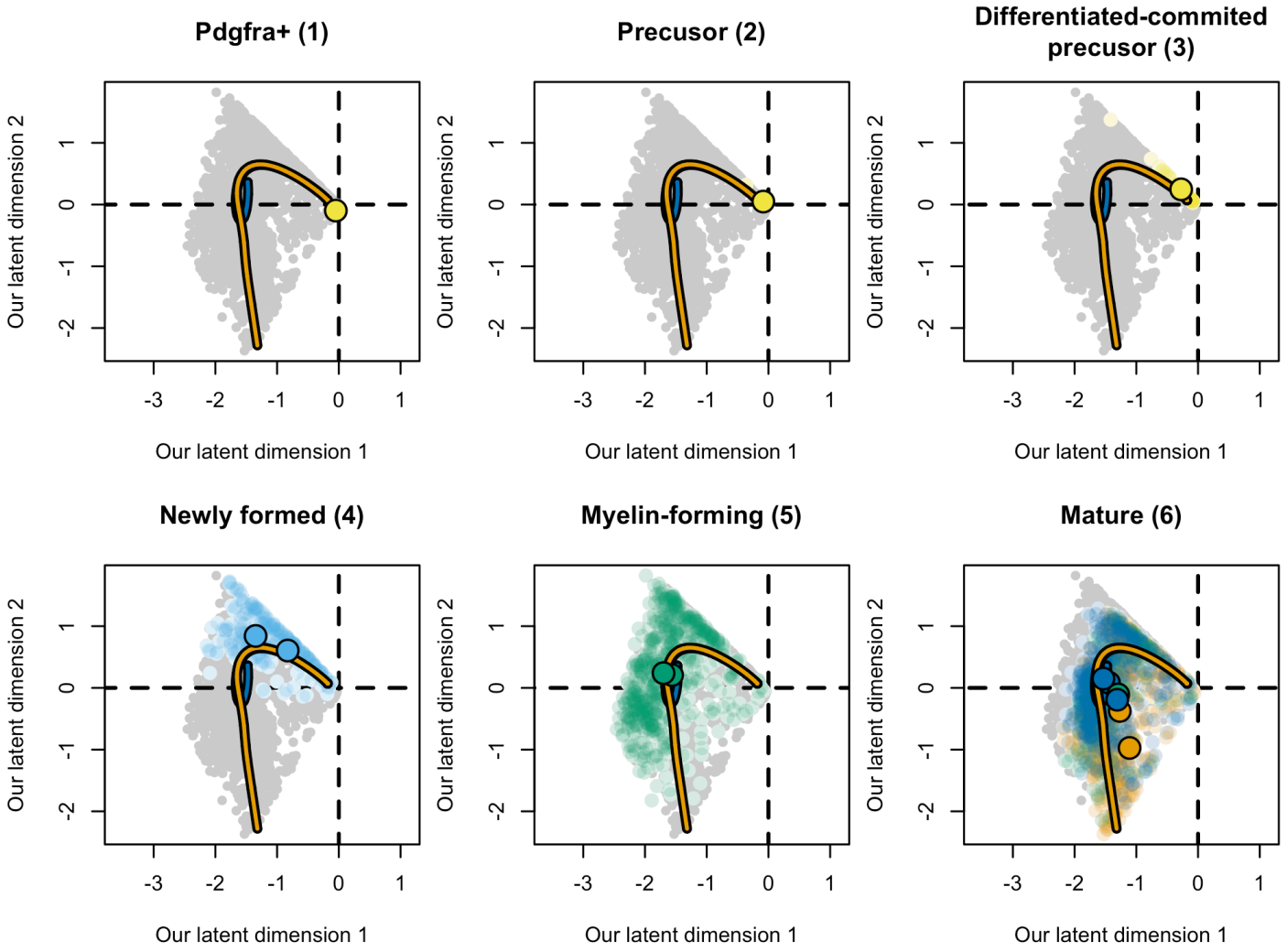
5. Estimating the trajectories

(Slingshot (Street et al., 2018))

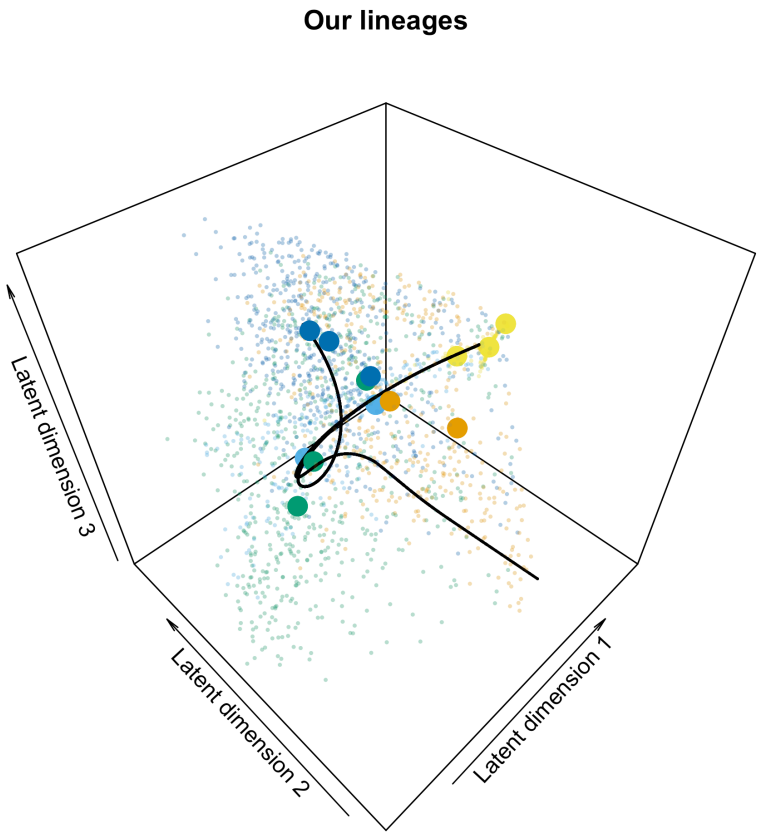
We estimate two trajectories in the oligodendrocytes, which diverge at the mature oligodendrocytes.

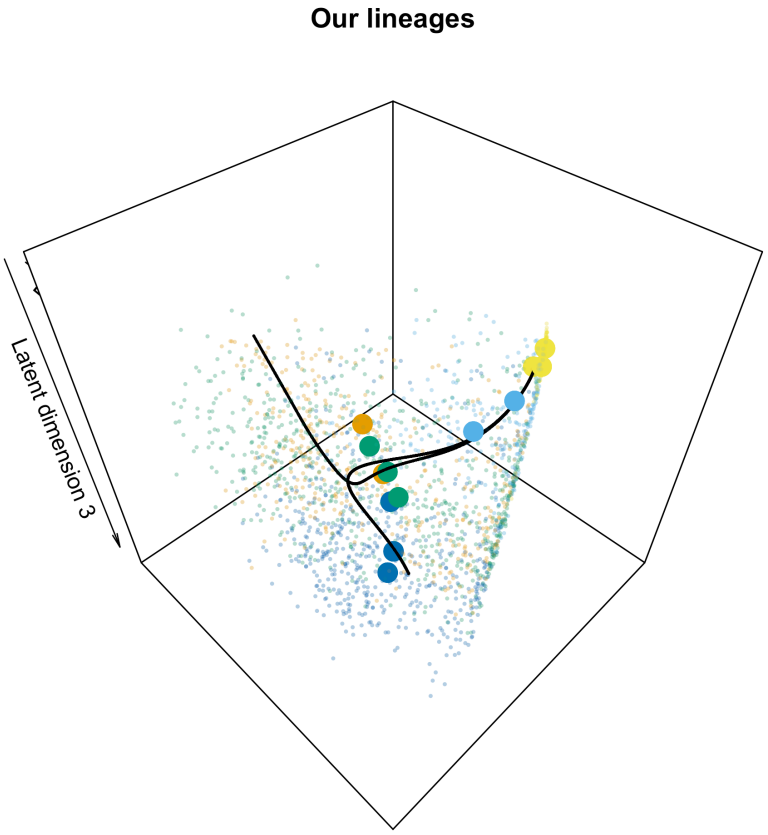
Major Cell Type (6 total)	Count (5000+ total)	Number of cell sub-types (13 total)
Pdgfra+ precursor	70+	1
Oligodendrocyte precursor cell	300+	1
Differentiation-committed oligodendrocyte precursors	100+	1
Newly formed oligodendrocytes	500+	2
Myelin-forming oligodendrocytes	1200+	2
Mature oligodendrocyte	2700+	6

We estimate two trajectories in the oligodendrocytes, which diverge at the mature oligodendrocytes.

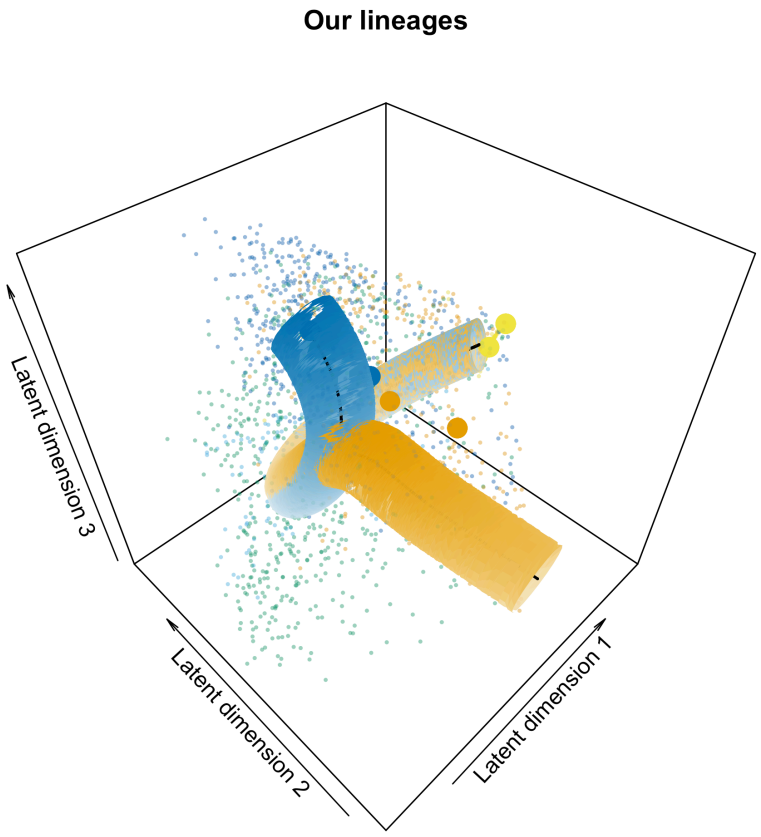


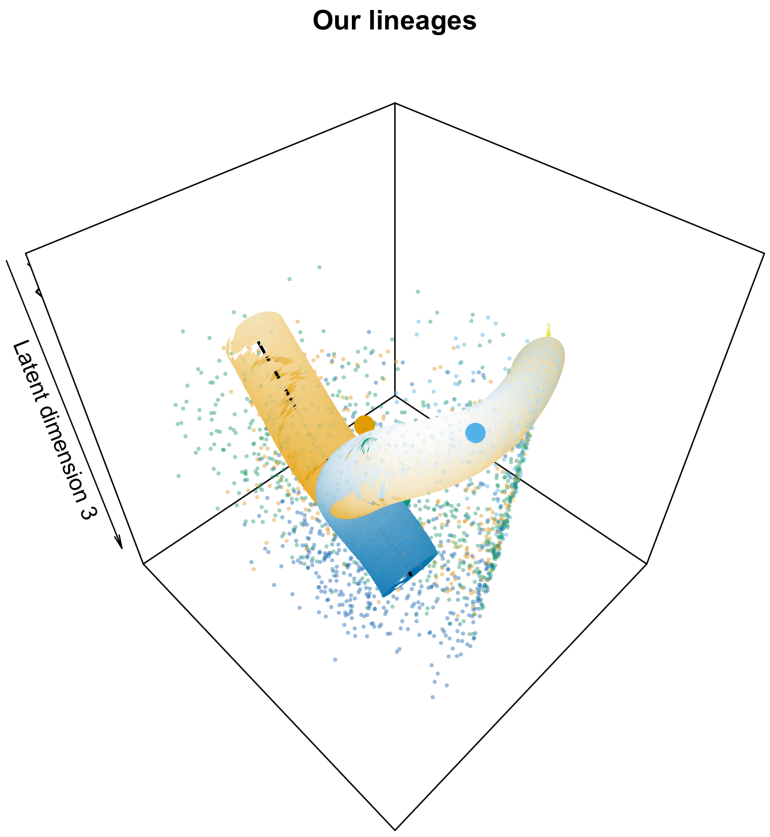
We estimate two trajectories in the oligodendrocytes, which diverge at the mature oligodendrocytes.



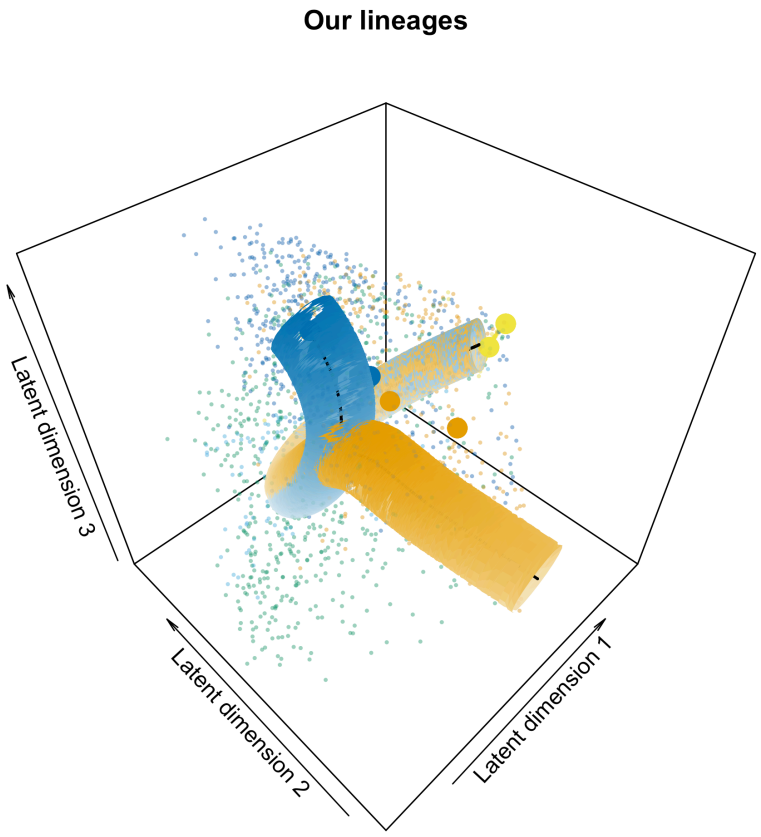


We use a bootstrap-based heuristic to build a “uncertainty tube” to see if the trajectories are indistinguishable.



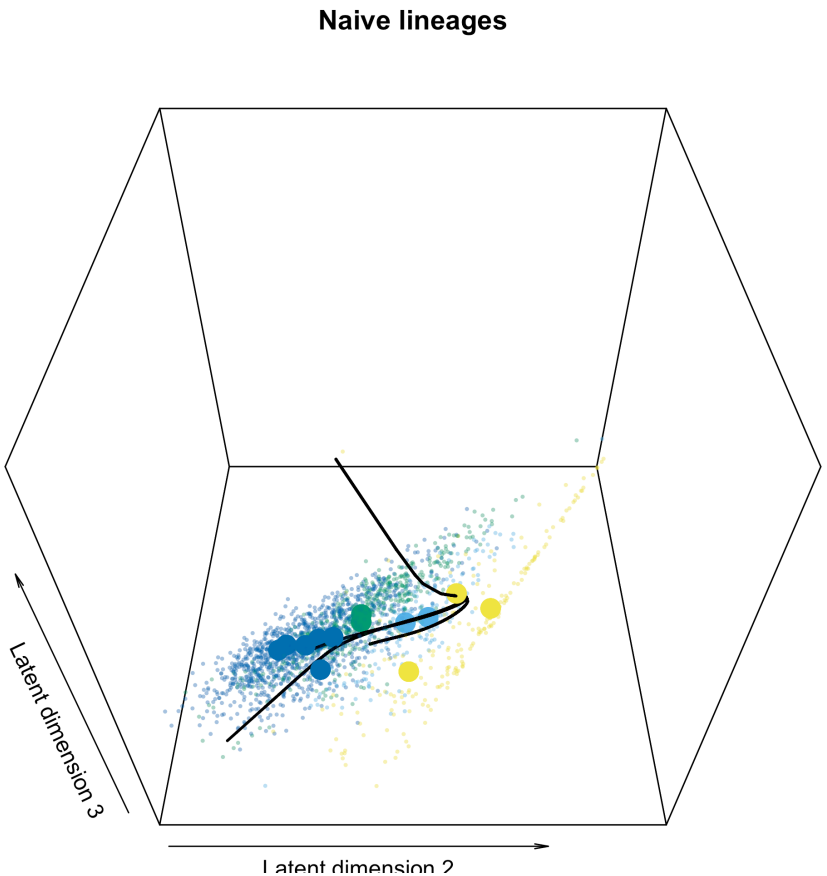


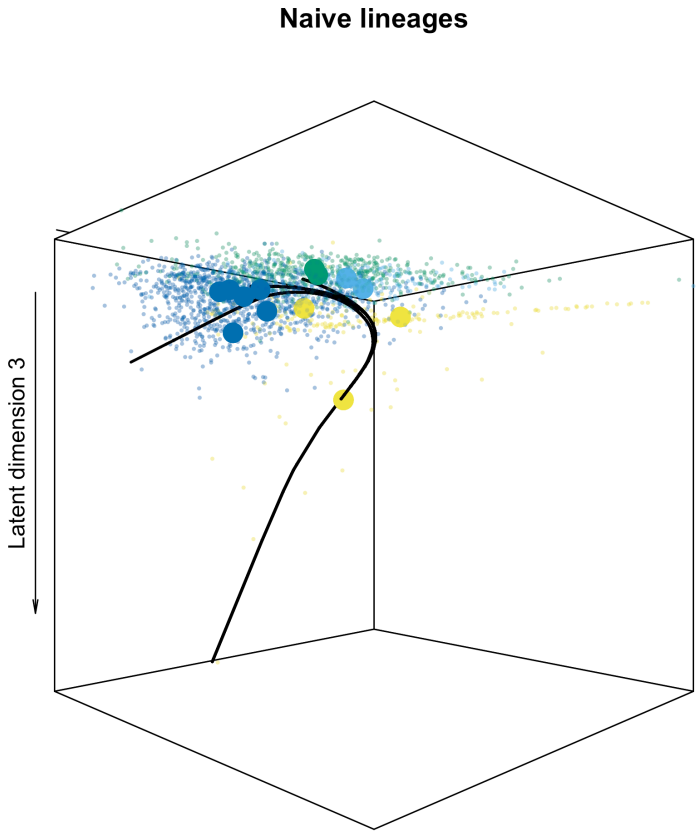
We use a bootstrap-based heuristic to build a “uncertainty tube” to see if the trajectories are indistinguishable.



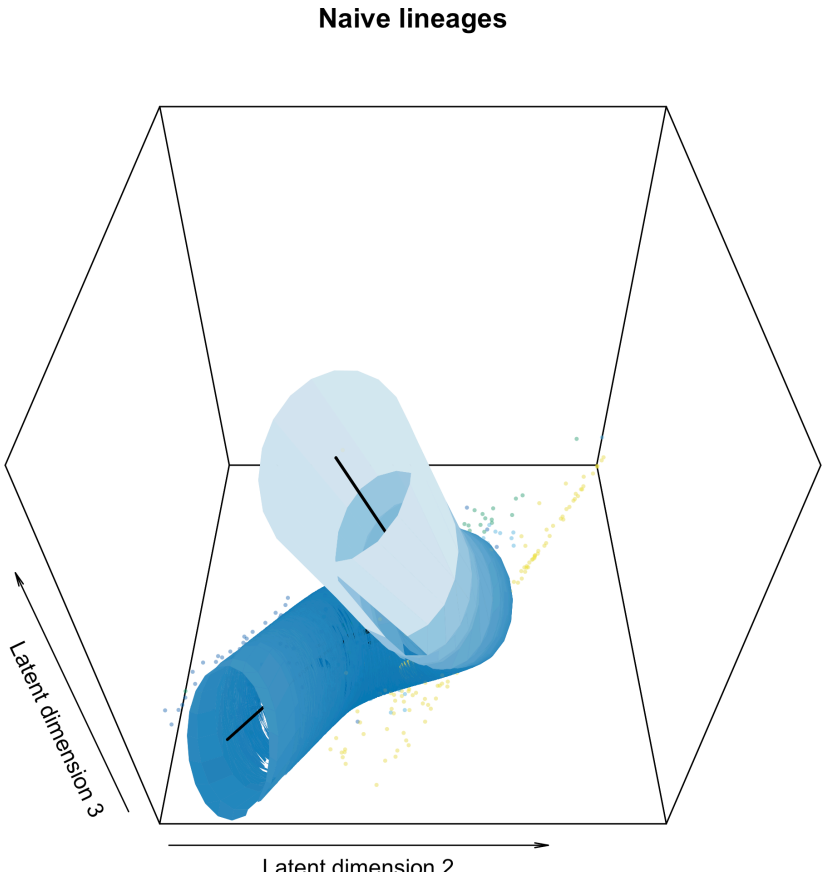
- Not a formal confidence set, but a useful visual diagnostic nonetheless
- Visible divergence of the two trajectories at the mature oligodendrocytes

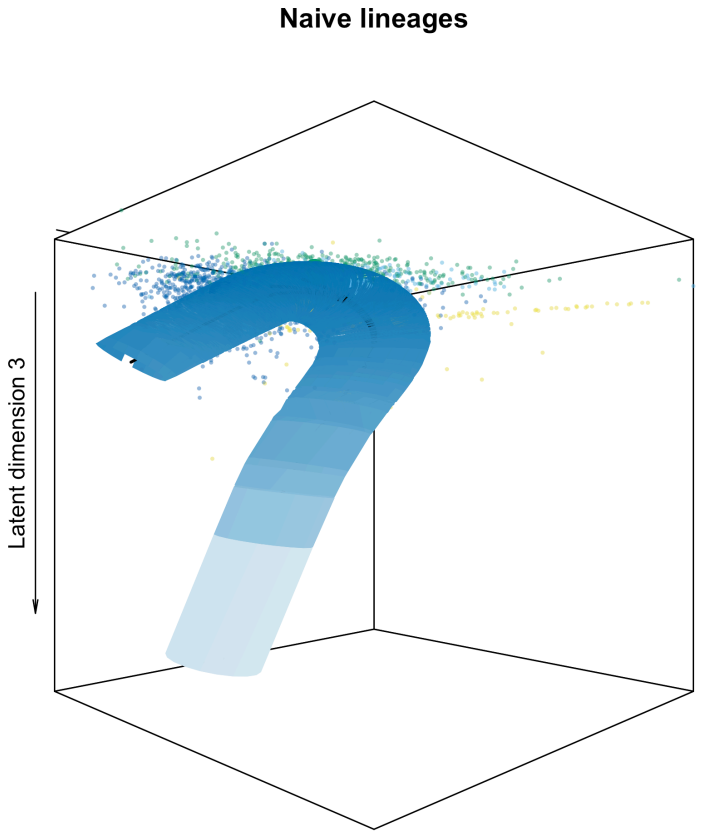
We use a bootstrap-based heuristic to build a “uncertainty tube” to see if the trajectories are indistinguishable.



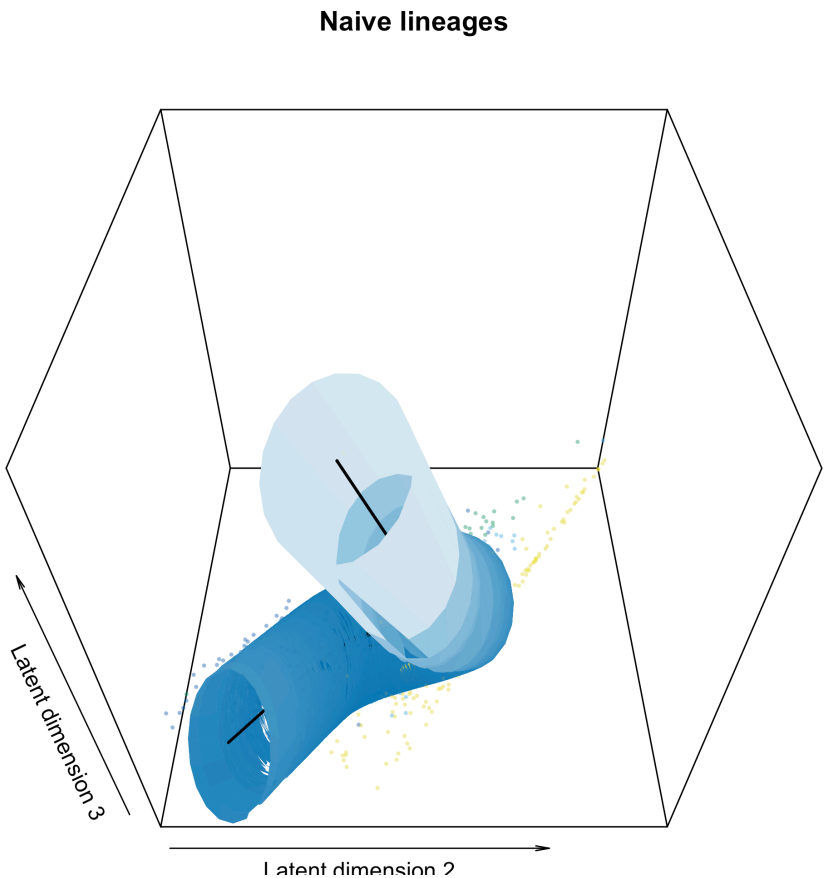


We use a bootstrap-based heuristic to build a “uncertainty tube” to see if the trajectories are indistinguishable.





We use a bootstrap-based heuristic to build a “uncertainty tube” to see if the trajectories are indistinguishable.



- All five trajectories are contained in the same uncertainty tube
- When using SVD (fixed-variance Gaussian), we recover similar results from previous work that suggest only one lineage

eSVD provides a general way to embed points to a lower-dimensional space for exponential-family distributions.

- We design eSVD, which embeds cells for an arbitrary exponential family distribution, and prove theory about the estimated embedding with respect to the hierarchical model.

eSVD provides a general way to embed points to a lower-dimensional space for exponential-family distributions.

- We design eSVD, which embeds cells for an arbitrary exponential family distribution, and prove theory about the estimated embedding with respect to the hierarchical model.

- We show that by using a curved Gaussian (where the standard deviation is half the mean) to analyze oligodendrocytes, we get a better model fit via diagnostics and observe two distinct trajectories that diverge at the mature oligodendrocytes.

eSVD provides a general way to embed points to a lower-dimensional space for exponential-family distributions.

- We design eSVD, which embeds cells for an arbitrary exponential family distribution, and prove theory about the estimated embedding with respect to the hierarchical model.

- We show that by using a curved Gaussian (where the standard deviation is half the mean) to analyze oligodendrocytes, we get a better model fit via diagnostics and observe two distinct trajectories that diverge at the mature oligodendrocytes.

Thank you!