scRNA-seq Dimension Reduction with scvis

Pascale Walters

2018-06-20

Introduction to scvis

```
library(scvis)
library(scater)
library(ggplot2)
library(cowplot)
```

`scvis` trains a probablistic parametric mapping for dimensionality reduction.

Example Data

We'll use an example `SingleCellExperiment` object from ascites data. This object has 500 cells and 5000 genes.

```
data("example_sce", package = "scvis")
example_sce
#> class: SingleCellExperiment
#> dim: 5000 500
#> metadata(1): log.exprs.offset
#> assays(1): logcounts
#> rownames(5000): A1BG A1BG-AS1 ... CTC-281F24.3 CTC-281F24.5
#> rowData names(0):
#> colnames(500): cell_1 cell_2 ... cell_622 cell_623
#> colData names(1): size_factor
#> reducedDimNames(0):
#> spikeNames(0):
```

The `logcounts` assay contains the expressions of each gene. We'll use this for dimensionality reduction. To minimize any noise in the data, we'll reduce to 100 dimensions.

```
example_sce <- runPCA(example_sce, ncomponents = 100)
```

Training the Model

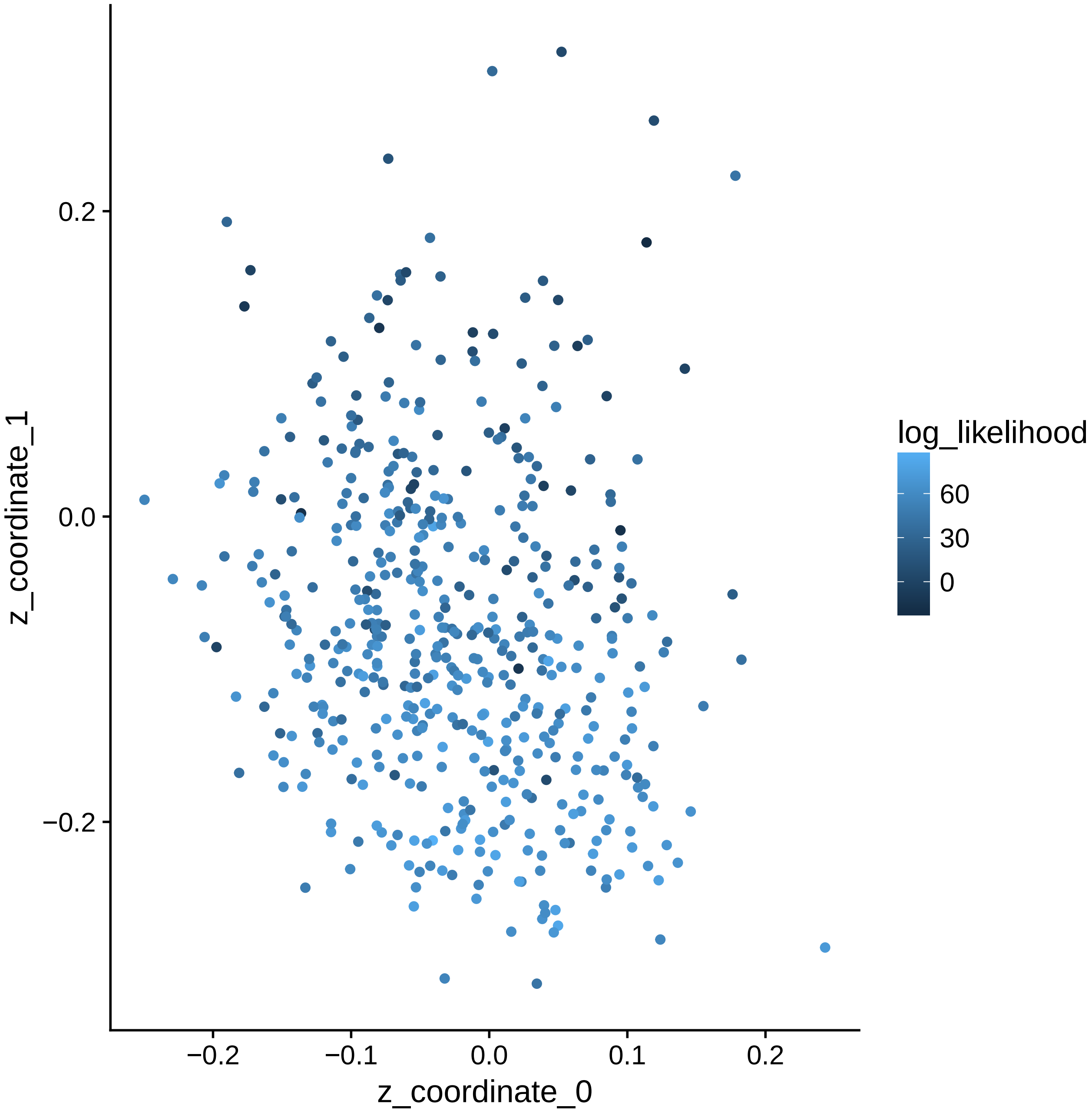
Now we can run scvis on the `SingleCellExperiment` object. To create a new embedding, we'll run `scvis_train` on `example_sce`.

```
scvis_sce <- scvis_train(example_sce, "output/", use_reducedDim = TRUE, reducedDim_name = "PCA")
```

This function performs dimension reduction on the results from `runPCA`. It writes the results to a .csv file in `output/`, as well as the log likelihoods, elbo results and t-SNE costs. The reduced dimensions are also added to the `scvis` slot in the `ReducedDimensions` of `example_sce`.

The trained model is saved in `output/model/`.

We can also plot the reduced dimensions:

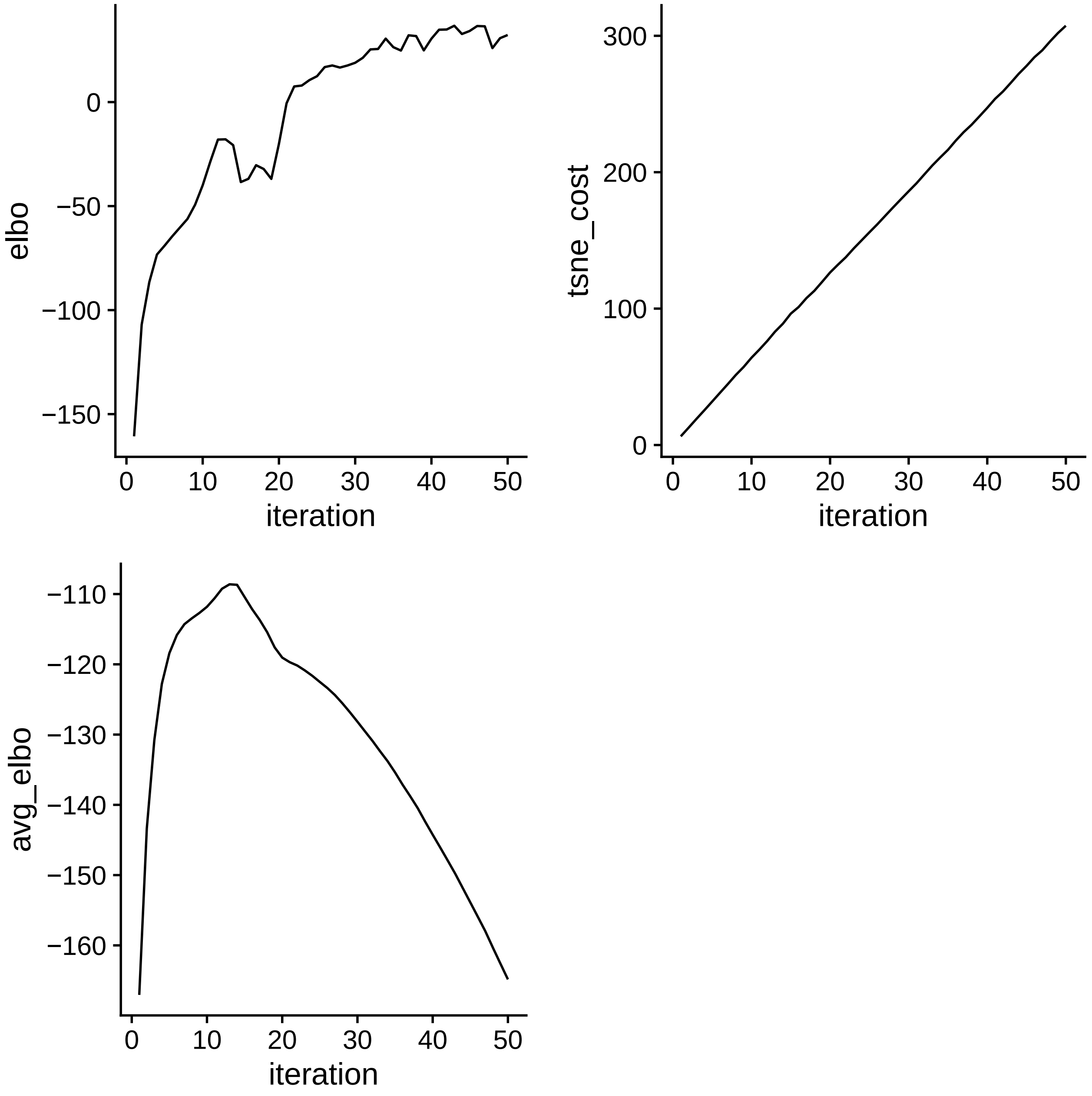


Finally, we can plot the elbo and t-SNE costs:

```
obj <- scvis_plot_obj_func("output/")

p1 <- ggplot(obj, aes(iteration, elbo)) + geom_line()
p2 <- ggplot(obj, aes(iteration, tsne_cost)) + geom_line()
p3 <- ggplot(obj, aes(iteration, avg_elbo)) + geom_line()

plot_grid(p1, p2, p3)
```



Adding New Data to an Existing Embedding

Suppose we had an existing trained model saved in `output/model/` and we wanted to add more data to it. We can use `scvis_map` with a `SingleCellExperiment` object.

The data that is added must have the same number of dimensions as the data used to train the model. For example, because we used data that had been reduced to 100 dimensions, we must use data in the `scvis_map` function that also has 100 dimensions.

```
output_dir <- file.path(find.package("scvis"), "vignette", "output")

scvis_map(example_sce, output_dir, use_reducedDim = TRUE, reducedDim_name = "PCA")
#> class: SingleCellExperiment
#> dim: 5000 500
#> metadata(1): log.exprs.offset
#> assays(1): logcounts
#> rownames(5000): A1BG A1BG-AS1 ... CTC-281F24.3 CTC-281F24.5
#> rowData names(0):
#> colnames(500): cell_1 cell_2 ... cell_622 cell_623
#> colData names(1): size_factor
#> reducedDimNames(2): PCA scvis
#> spikeNames(0):
```

This adds the data from the `counts` assay of `example_sce` to the model that is stored in `output/model/`. It writes the results to a .csv file in `output/`, as well as the log likelihoods. The reduced dimensions are also added to the `scvis` slot in the `ReducedDimensions` of `example_sce`.