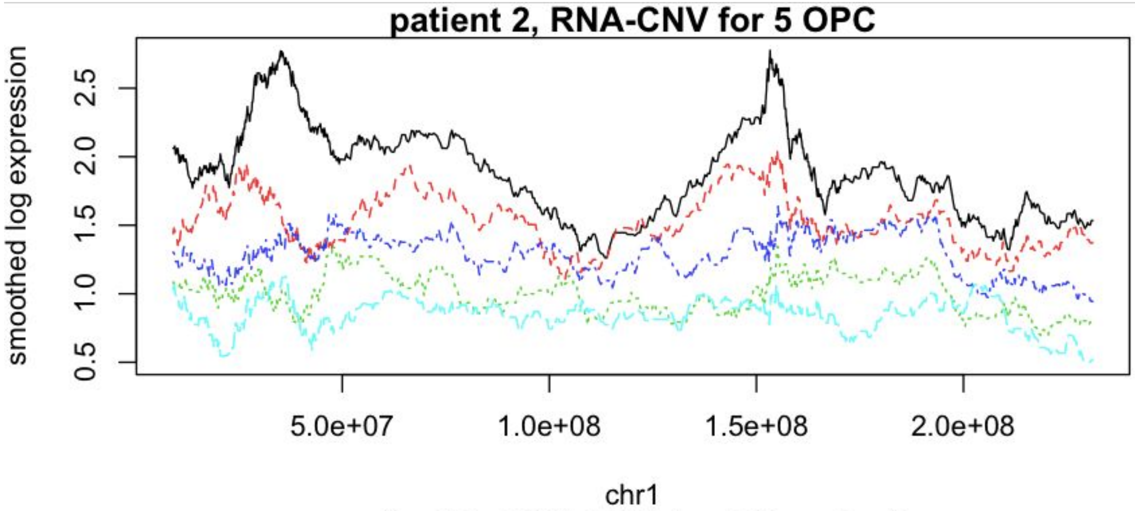
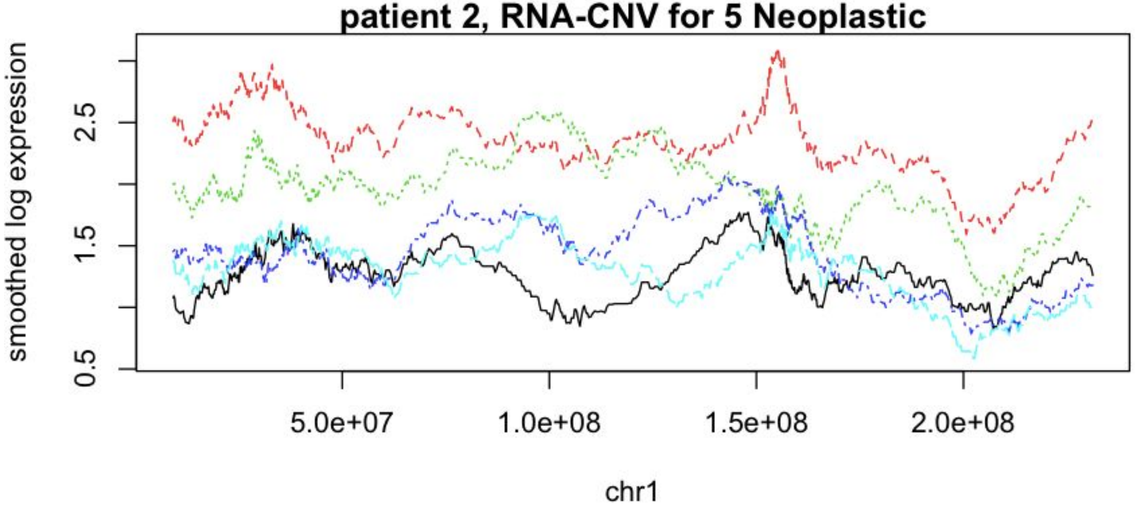
Approaching mastery of distance, dimension reduction, clustering and classification in genomic applications

VJ Carey, PhD
CSAMA 2019
Bressanone IT

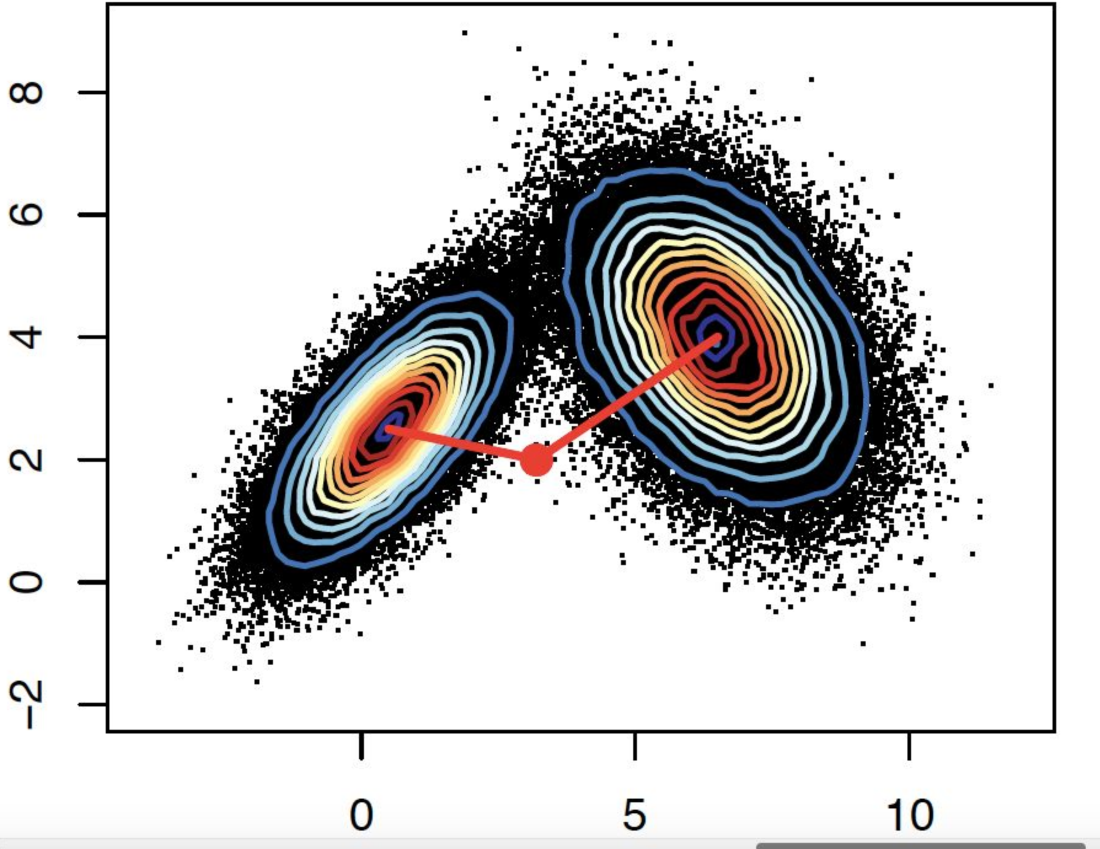

patient 2, RNA-CNV for 5 OPC


patient 2, RNA-CNV for 5 Neoplastic

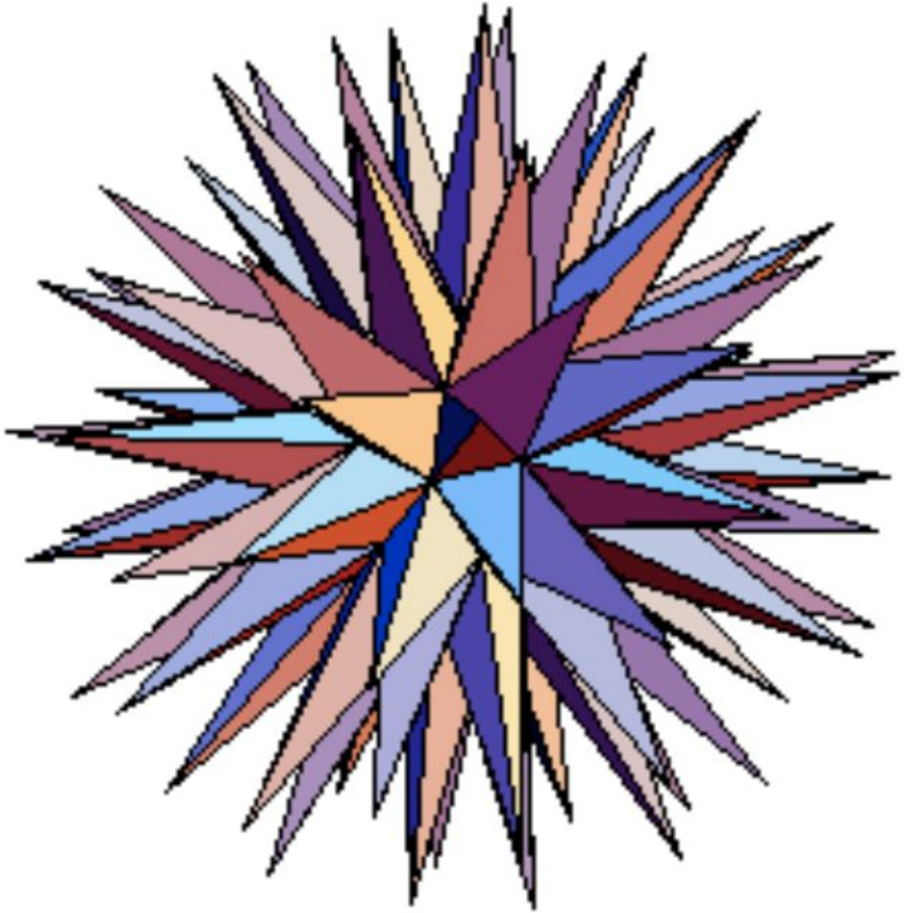
- x axis: genomic coordinate on chr1
- y axis: expression smoothed over windows of ~100 genes
- OPC: oligodendrocyte precursor cells
- What approach to measuring cell-cell distance should be used?
- How would you go about feature selection for classifying cells?

Mastery:

- n. comprehensive knowledge or skill in a particular subject or activity
- mastery of distances?
- From Holmes and Huber MSMB: to which cluster center is the red dot closest?



- Mathematics of the 19th century: e minimize the role of geometric intuition (Dedekind, Hilbert)
- Is it reasonable to make use of spatial intuition in biology?



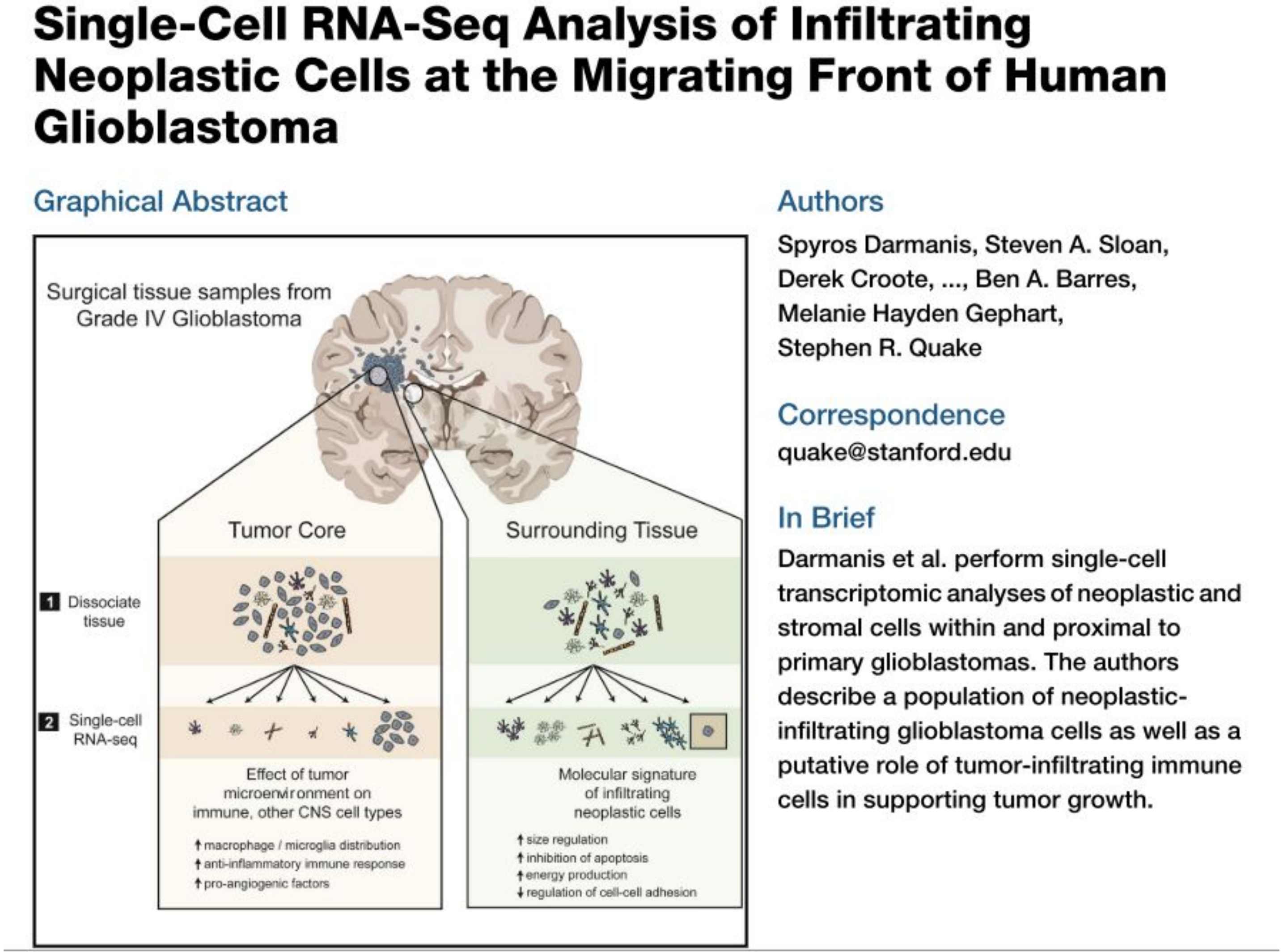
An attempt to visualize a 7-dim hypercube ($2^7 = 128$ corners)

http://yaroslavvb.blogspot.com/2006/05/curse-of-dimensionality-and-intuition.html

Road map

- Case study: single-cell RNA-seq in glioblastoma
- Distances and the curse of dimensionality
- Dimension reduction and feature engineering
- Options and figures of merit in cluster analysis
- Concepts of supervised learning
- kipoi.org: an archive of trained models

Cell Reports, 2017



Single-Cell RNA-Seq Analysis of Infiltrating Neoplastic Cells at the Migrating Front of Human Glioblastoma

Graphical Abstract

Surgical tissue samples from Grade IV Glioblastoma

Tumor Core

Surrounding Tissue

1 Dissociate tissue

2 Single-cell RNA-seq

Effect of tumor microenvironment on immune, other CNS cell types

↑ macrophage / microglia distribution
↑ anti-inflammatory immune response
↑ pro-angiogenic factors

Molecular signature of infiltrating neoplastic cells

↑ size regulation
↑ inhibition of apoptosis
↑ energy production
↓ regulation of cell-cell adhesion

Authors

Spyros Darmanis, Steven A. Sloan, Derek Croote, ..., Ben A. Barres, Melanie Hayden Gephart, Stephen R. Quake

Correspondence

quake@stanford.edu

In Brief

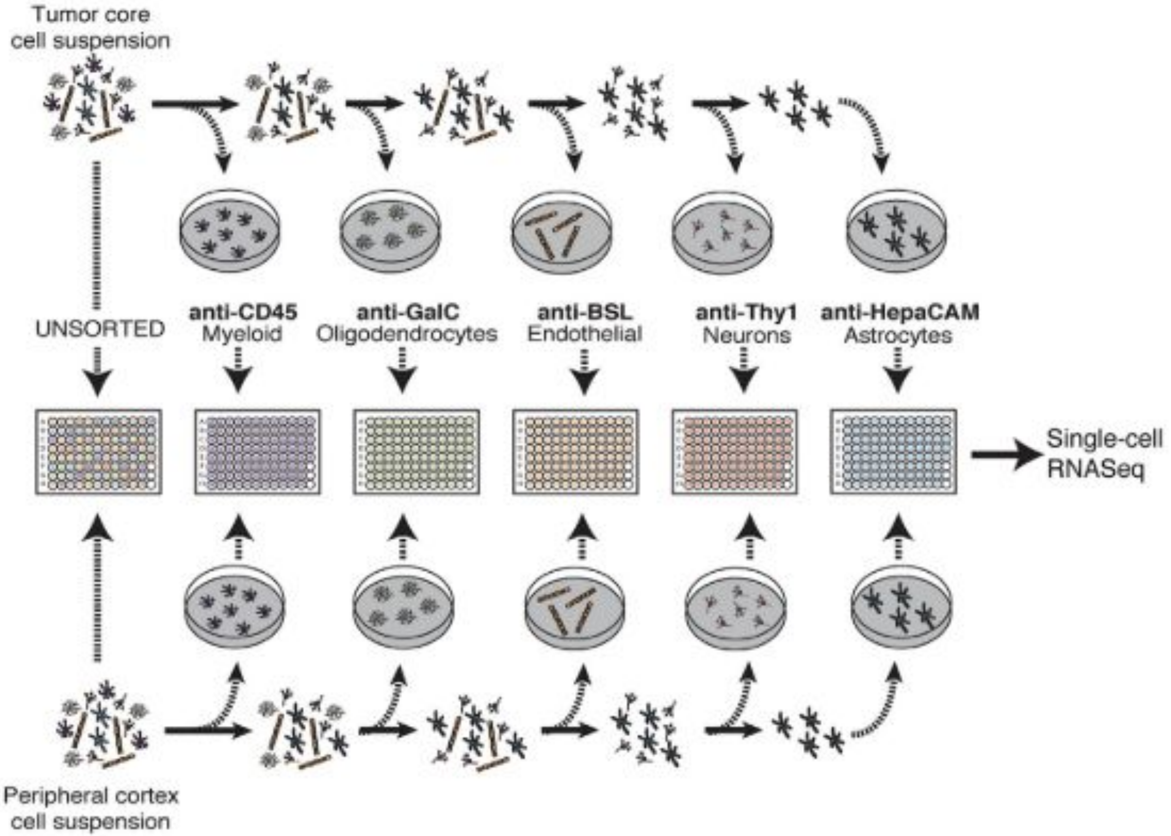
Darmanis et al. perform single-cell transcriptomic analyses of neoplastic and stromal cells within and proximal to primary glioblastomas. The authors describe a population of neoplastic-infiltrating glioblastoma cells as well as a putative role of tumor-infiltrating immune cells in supporting tumor growth.

Design summary

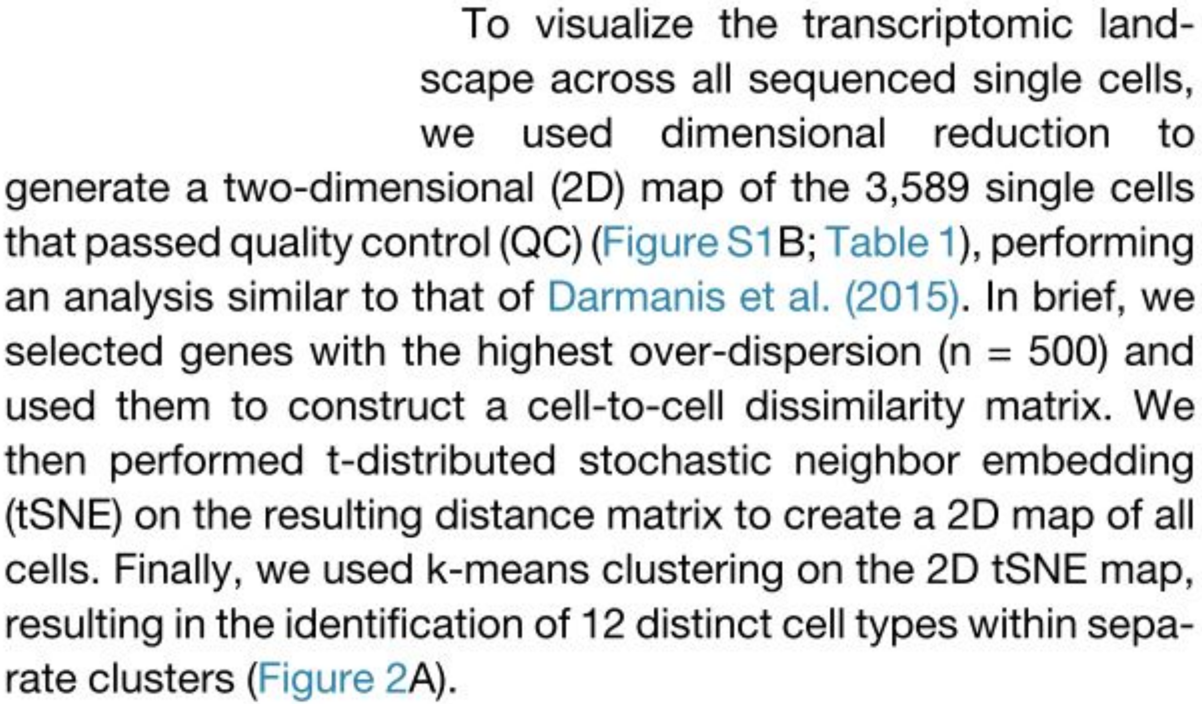
- 3500 cells from glioblastoma samples from four patients (IDH1-negative)
- Cells isolated from tumor core and periphery, immunopanned to increase diversity of cell types
- Smart-seq2 scRNA-seq on all cells
    - t-SNE+k-means used to identify 12 clusters
    - biological identity of clusters inferred via signature assessment
    - smoothing of expression profiles used to obtain CNV profiles
    - hierarchical clustering of CNV profiles exposes distinctions of neoplastic and non-neoplastic cells
    - differential expression to obtain signature of infiltrating cells

Cell selection via immunopanning

Goal: "encompassing the entirety of the tumor and peritumor cellular landscape that is often
blurred in bulk sequencing studies or insufficiently sampled in prior single cell studies"



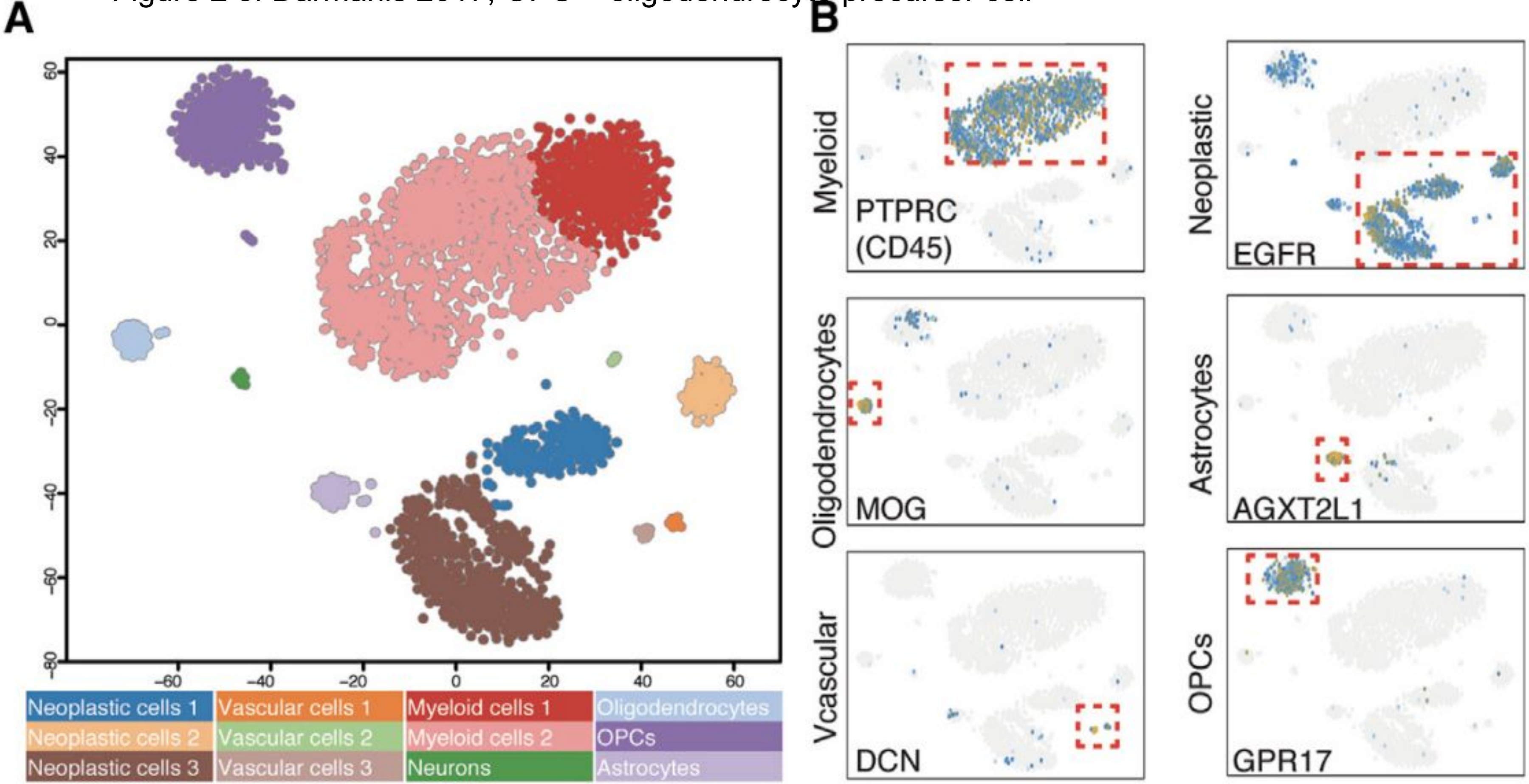
Setup for dimension reduction (Darmanis 2017 Figure 2)



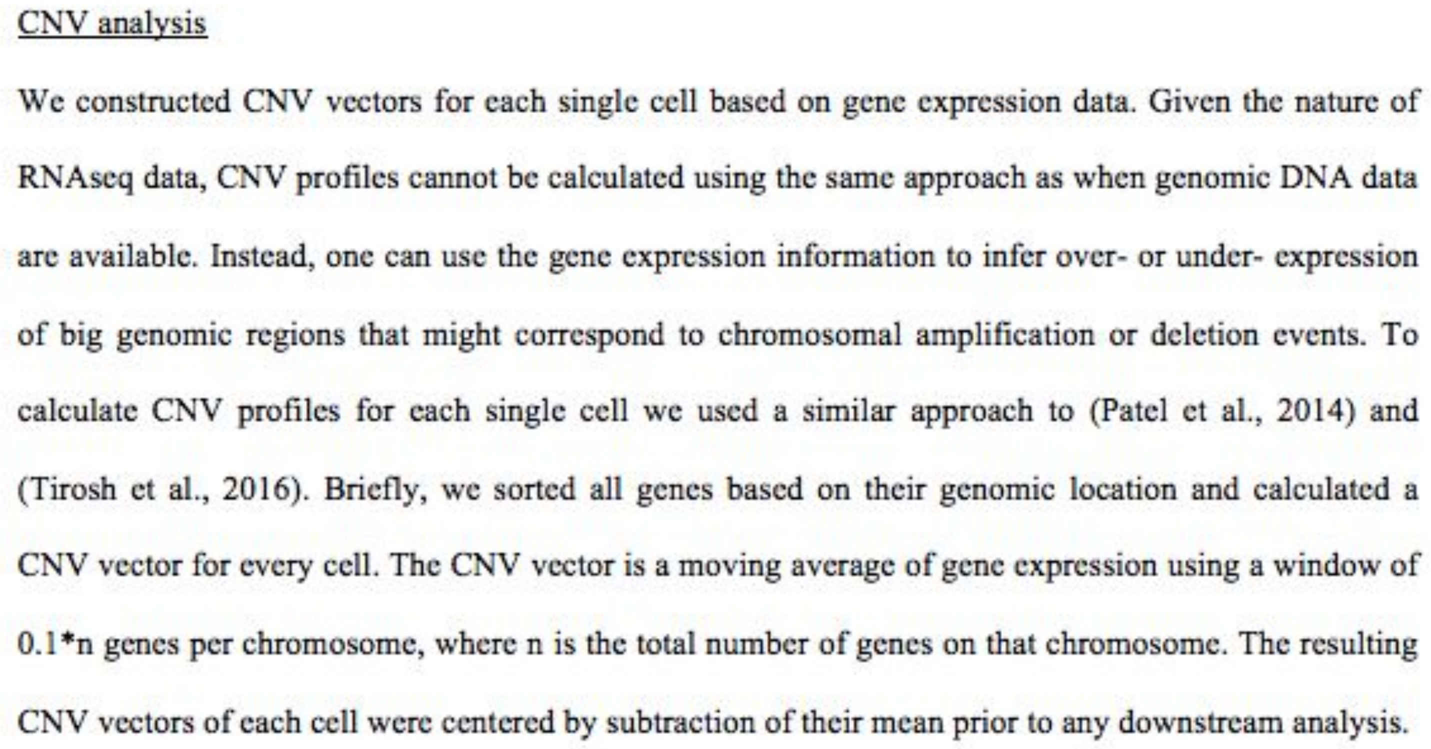
To visualize the transcriptomic landscape across all sequenced single cells, we used dimensional reduction to generate a two-dimensional (2D) map of the 3,589 single cells that passed quality control (QC) (Figure S1B; Table 1), performing an analysis similar to that of Darmanis et al. (2015). In brief, we selected genes with the highest over-dispersion (n = 500) and used them to construct a cell-to-cell dissimilarity matrix. We then performed t-distributed stochastic neighbor embedding (tSNE) on the resulting distance matrix to create a 2D map of all cells. Finally, we used k-means clustering on the 2D tSNE map, resulting in the identification of 12 distinct cell types within separate clusters (Figure 2A).

- Feature selection via PAGODA pathway-oriented overdispersion metric
- dissimilarity metric is d(x,y) = 1- cor(x,y) where x and y are vectors of expression measures over all samples
- t-SNE perplexity set to 50

Figure 2 of Darmanis 2017; OPC = oligodendrocyte precursor cell

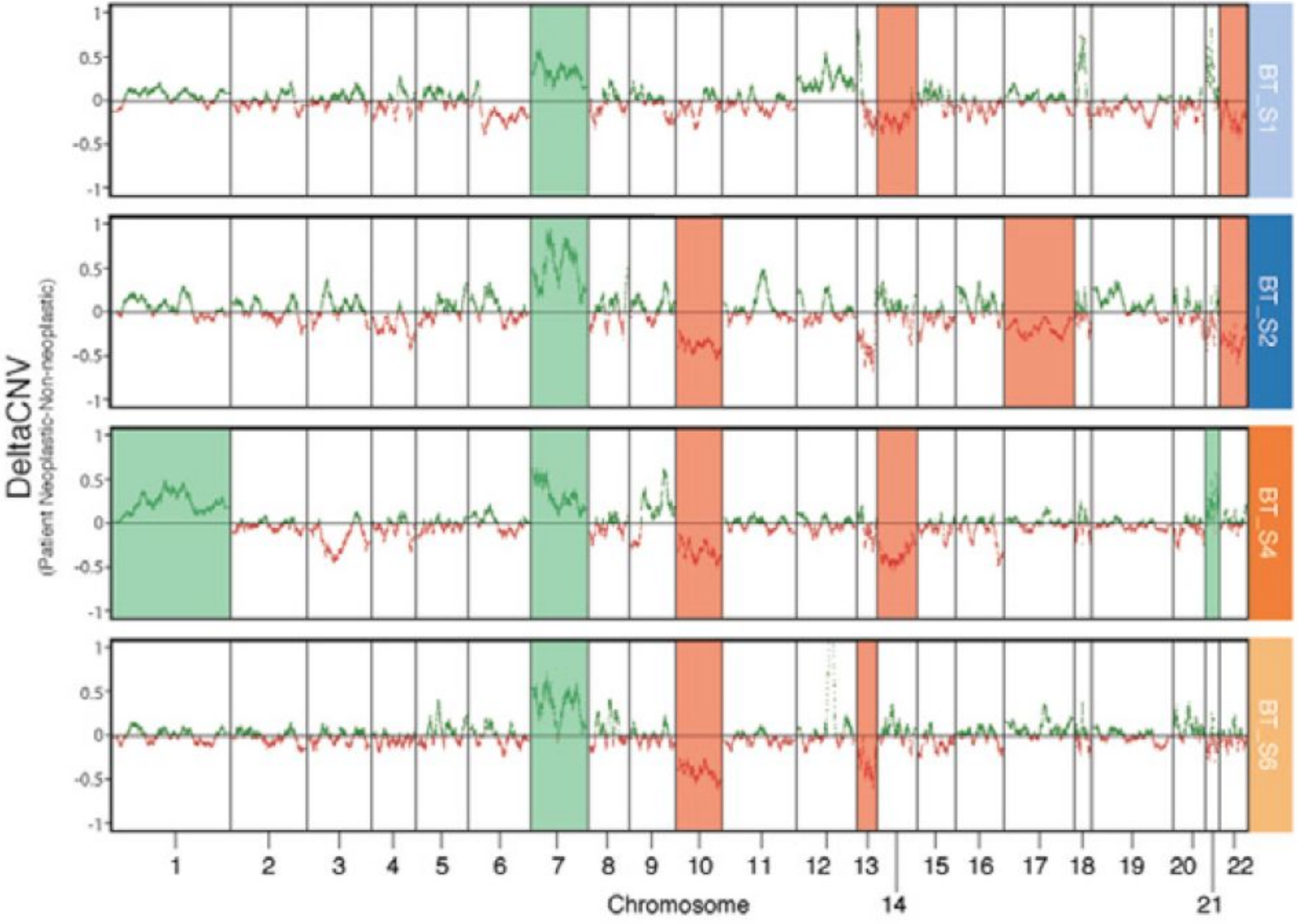


Single-cell CNV profiling via single-cell RNA-seq [supplement]

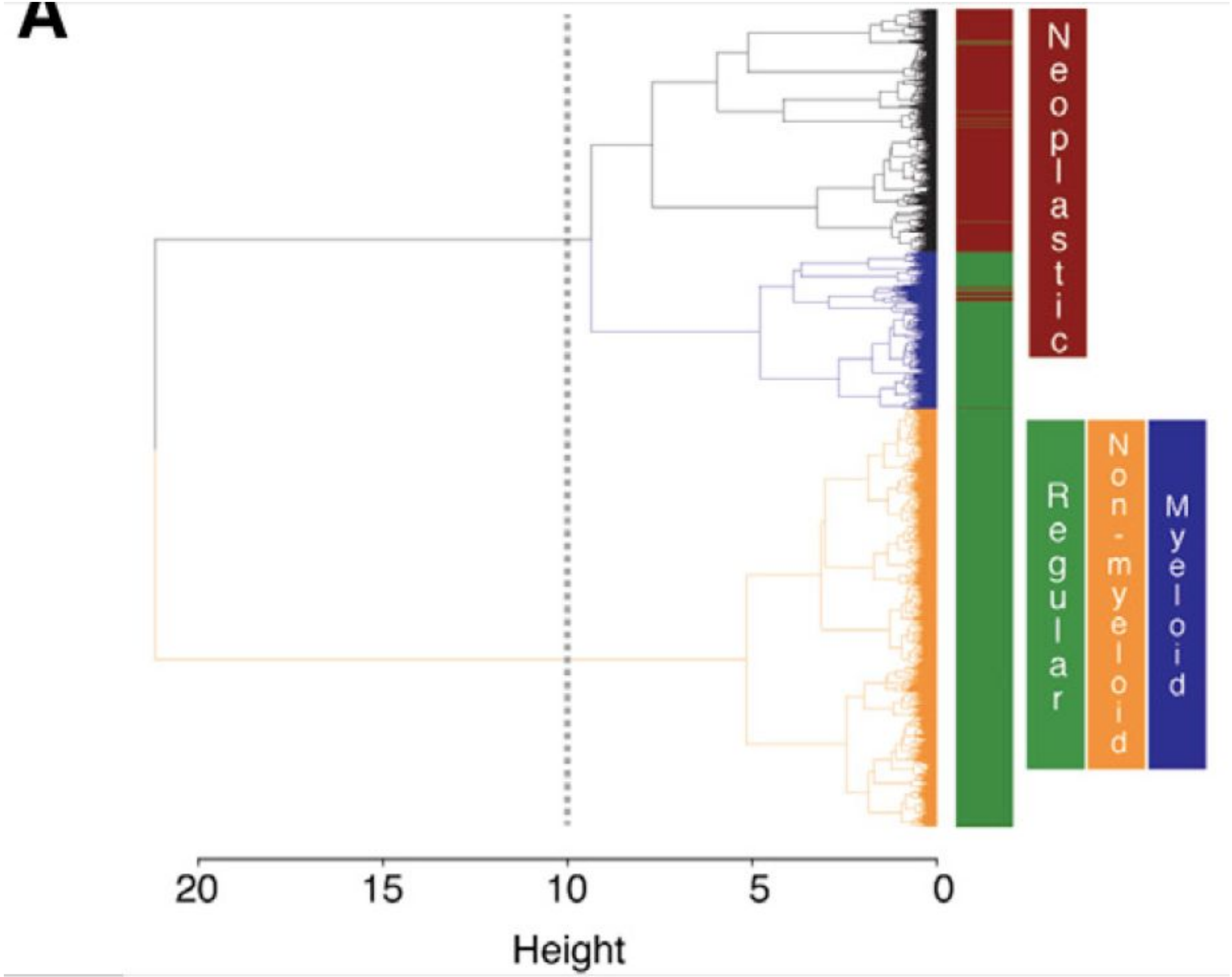


CNV analysis

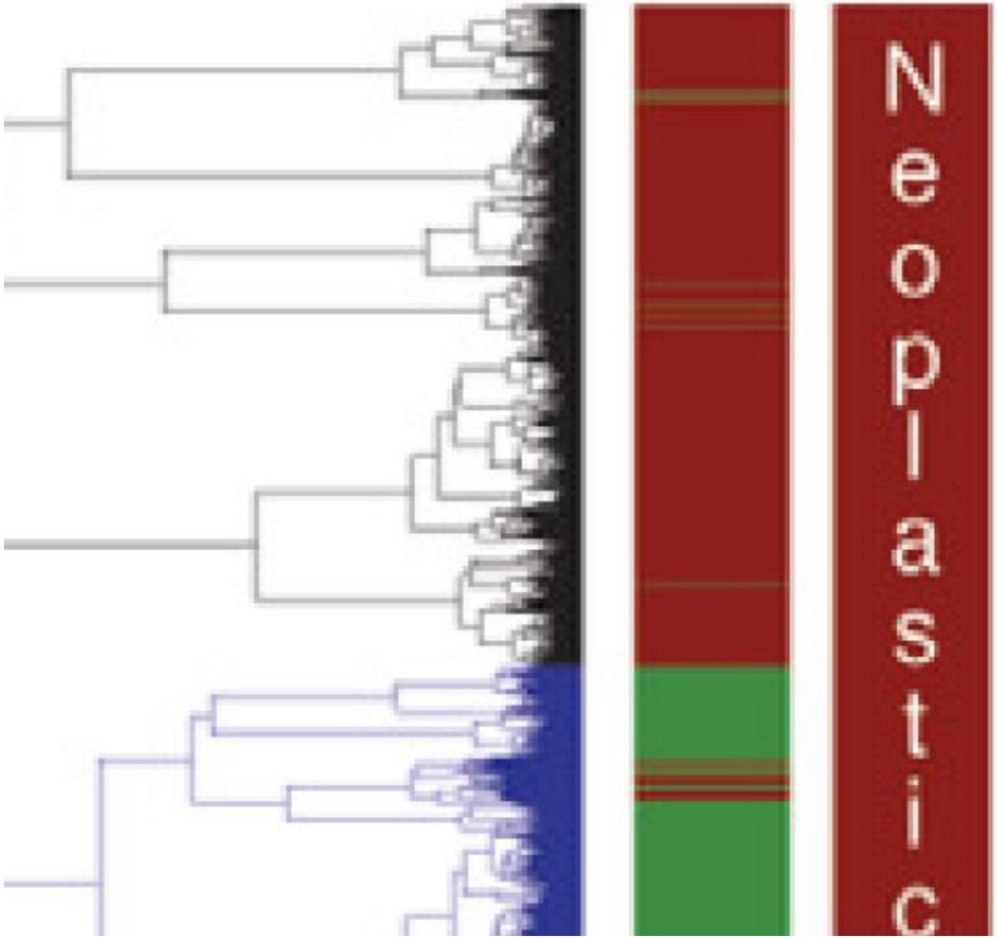
We constructed CNV vectors for each single cell based on gene expression data. Given the nature of RNAseq data, CNV profiles cannot be calculated using the same approach as when genomic DNA data are available. Instead, one can use the gene expression information to infer over- or under- expression of big genomic regions that might correspond to chromosomal amplification or deletion events. To calculate CNV profiles for each single cell we used a similar approach to (Patel et al., 2014) and (Tirosh et al., 2016). Briefly, we sorted all genes based on their genomic location and calculated a CNV vector for every cell. The CNV vector is a moving average of gene expression using a window of 0.1*n genes per chromosome, where n is the total number of genes on that chromosome. The resulting CNV vectors of each cell were centered by subtraction of their mean prior to any downstream analysis.



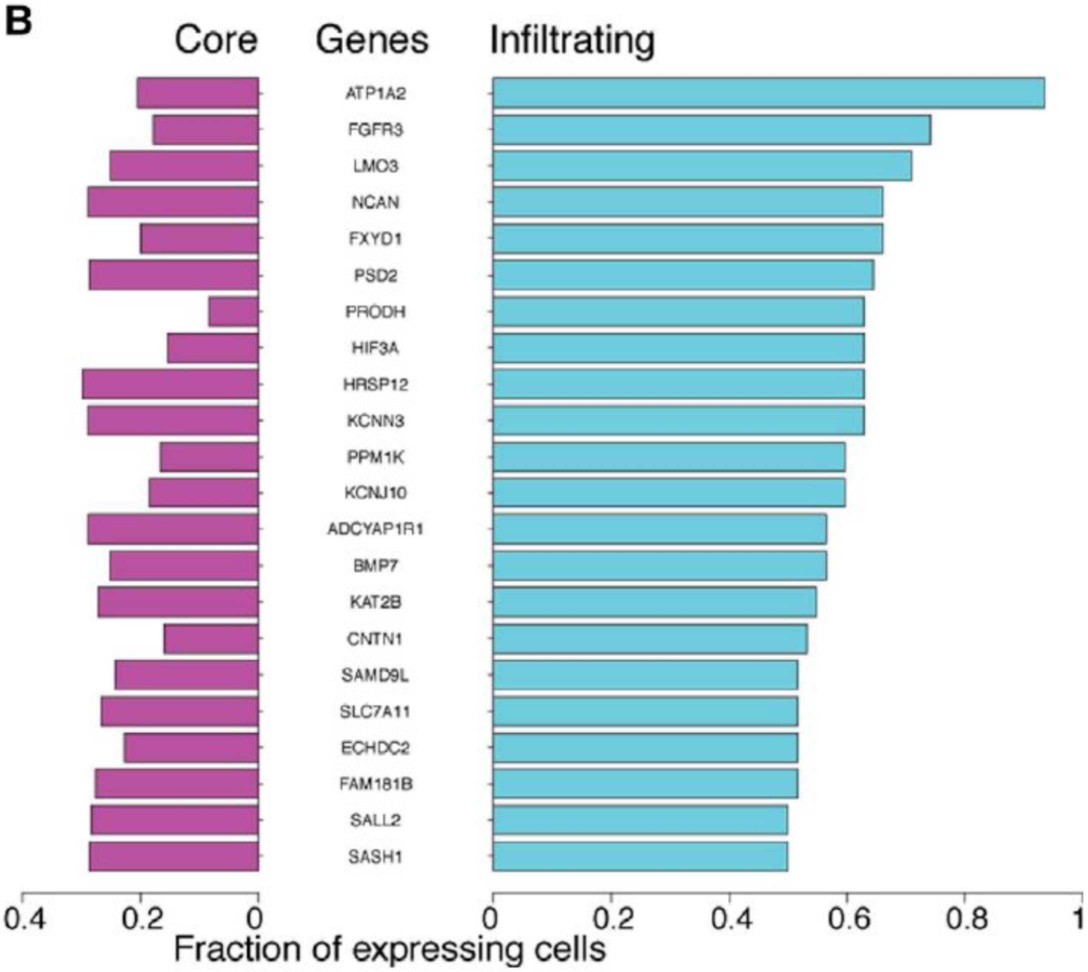
- Four patients
- RNA-seq profile for each cell is smoothed
- For each patient, average for non-neoplastic cells is subtracted from average for neoplastic cells
- The cell-specific CNV profiles are used to form a measure of structural (as opposed to transcriptomic) distance between cells, for hierarchical clustering



- "The resulting dendrogram was composed of three primary branches (Figure 3A): one (CNV 1) consisted exclusively of neoplastic cells, whereas the remaining two contained the majority of non-neoplastic cells."
- Details of hierarchical clustering not discussed
- Options include the form of distance/dissimilarity, method of agglomeration, criterion of labeling (cutting the tree)



- Upon magnification, it appears that there are numerous non-neoplastic cells (green bars) in the branch colored black or brown
- Authors report low misclassification rates, and there are other approaches to confirming the plausibility of the CNV profiling reported in the paper



Major result:
A set of genes characteristic of the infiltrating cells (boundary of tumor)

DESeq2 declared 1000/250 genes down/upregulated comparing peripheral to core

This list involves genes expressed in more than 50% of infiltrators but fewer than 30% of core

Where do "50%" and "30%" come from?

Summary

- t-SNE dimension reduction leads to groupings of cells that can be rationalized in terms of brain and tumor anatomy
- k-means clustering was used in the 2-d space
- smoothing expression vectors over genomic coordinates leads to RNA-seq based CNV profiles for each cell
- hierarchical clustering was used with these CNV profiles, and distinguished groups of neoplastic and non-neoplastic cells
- cells on tumor periphery have a distinct expression signature that is rationalized by GO categories, etc.

Questions

- Is a reduction to two dimensions sufficient for what we want to do?
- Should we consider alternatives to the distance d(x,y) = (1-cor(x,y)) underlying the t-SNE rendering in the paper?
- Should we consider alternatives to t-SNE for dimension reduction? Is tuning (e.g., setting of perplexity and "learning rate") worth exploring?
- Is there a tuning aspect of the hierarchical clustering of cell-specific CNV profiles worth exploring?

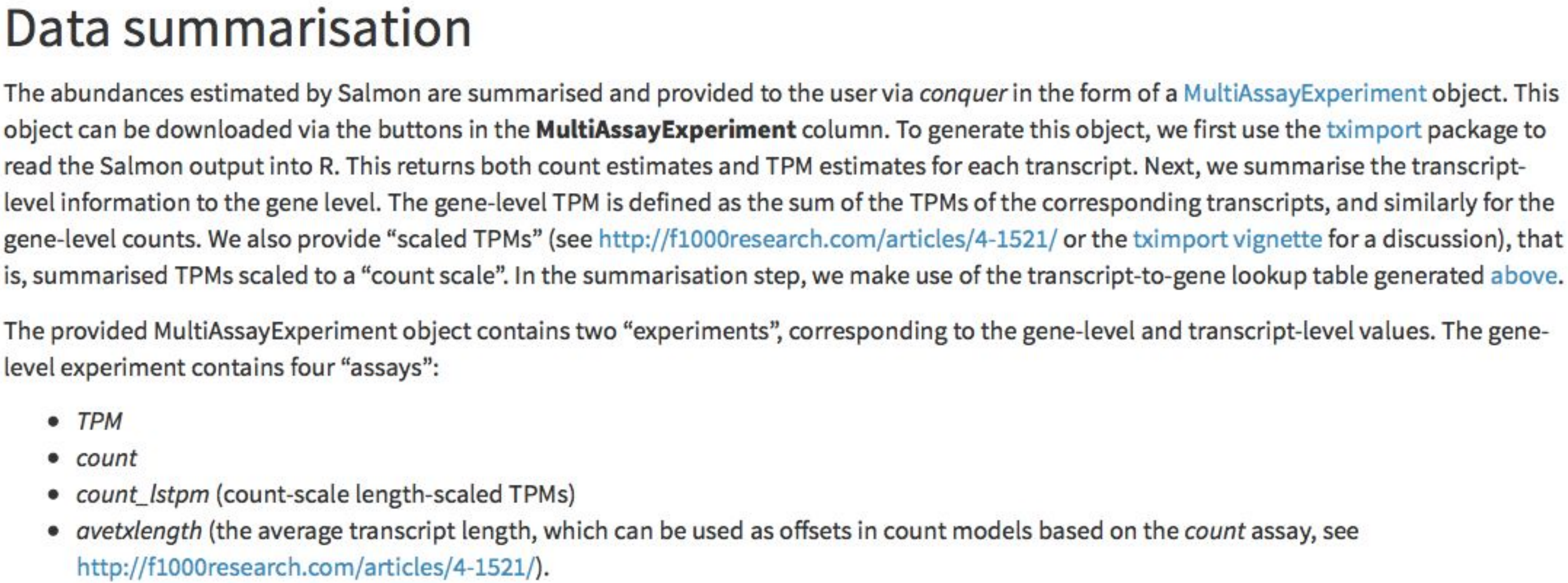
To start to address these questions, we will start to work with the Darmanis data in a certain structure produced by Charlotte Soneson in the CONQUER [consistent quantification for RNA-seq data] system

Comments

- I don't know the exact set of 500 genes used by Darmanis, as they were identified using PAGODA's overdispersion metric, so I select ~700 genes ordered by overall s.d. across all samples (omitting some with very large s.d.s that disrupt reasonable visualization strategies)
- The result of Rtsne (code to be shown) with minimal tuning recapitulates aspects of the Darmanis published display, and constitutes a sanity check for the various tasks of deriving and analyzing the data separately from the authors
- I use the GEO-based labeling of cells -- I do not have the classes asserted in the published figures
- We can now explore sensitivity of the t-SNE procedure to tuning parameter selection
- We can now explore effects of choosing other dimension reduction approaches for this analysis task

RNA-seq quantifications: I use "count-scale length-scaled TPM"

from conquer "about" tab:



Data summarisation

The abundances estimated by Salmon are summarised and provided to the user via *conquer* in the form of a MultiAssayExperiment object. This object can be downloaded via the buttons in the **MultiAssayExperiment** column. To generate this object, we first use the tximport package to read the Salmon output into R. This returns both count estimates and TPM estimates for each transcript. Next, we summarise the transcript-level information to the gene level. The gene-level TPM is defined as the sum of the TPMs of the corresponding transcripts, and similarly for the gene-level counts. We also provide "scaled TPMs" (see http://f1000research.com/articles/4-1521/ or the tximport vignette for a discussion), that is, summarised TPMs scaled to a "count scale". In the summarisation step, we make use of the transcript-to-gene lookup table generated above.

The provided MultiAssayExperiment object contains two "experiments", corresponding to the gene-level and transcript-level values. The gene-level experiment contains four "assays":

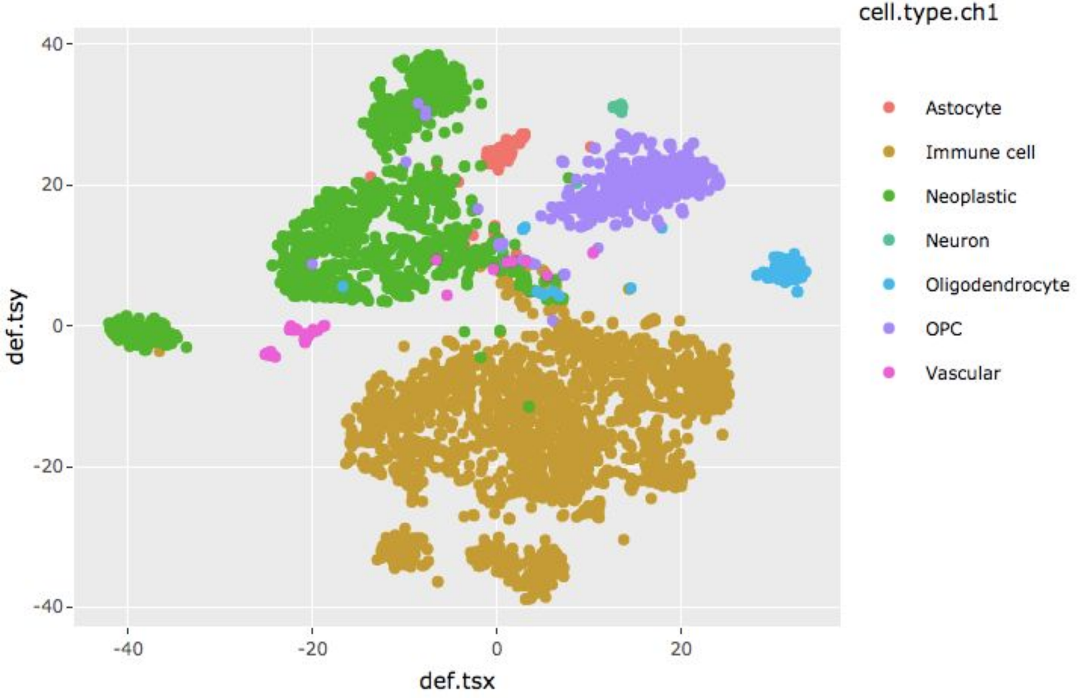
- *TPM*
- *count*
- *count_lstpm* (count-scale length-scaled TPMs)
- *avetxlength* (the average transcript length, which can be used as offsets in count models based on the *count* assay, see http://f1000research.com/articles/4-1521/).

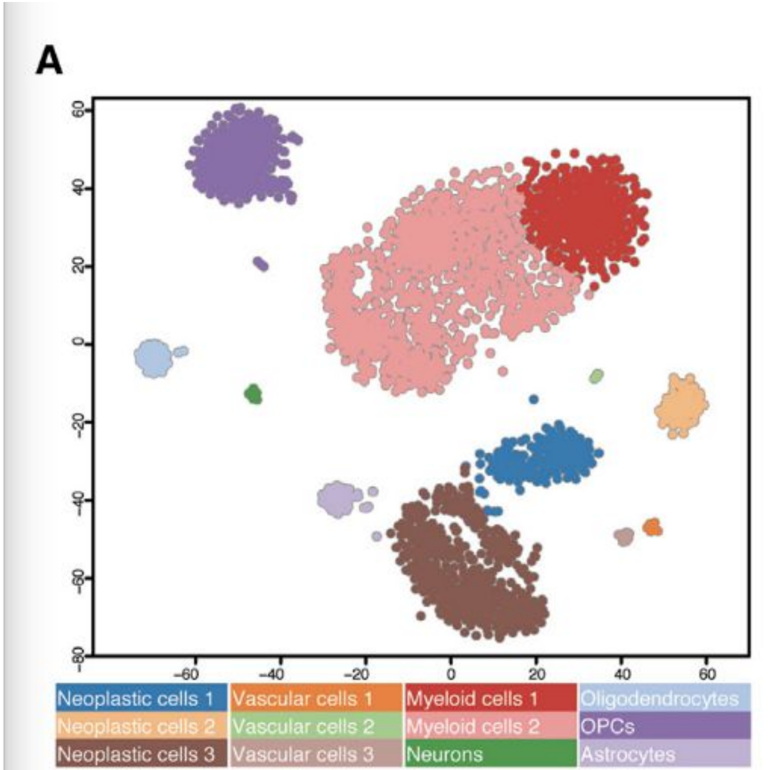
A basic representation of the Darmanis 2017 data after extraction from CONQUER

```
> locdarm
class: RangedSummarizedExperiment
dim: 65218 3584
metadata(0):
assays(1): count_lstpm
rownames(65218): ENSG00000000003.14 ENSG00000000005.5 ... ERCC-00170
  ERCC-00171
rowData names(3): gene genome symbol
colnames(3584): GSM2243439 GSM2243440 ... GSM2247076 GSM2247077
colData names(59): title geo_accession ... tsne.cluster.ch1 well.ch1
```

Reduction to ~700 genes using s.d. over all samples is elementary … rowSds and [

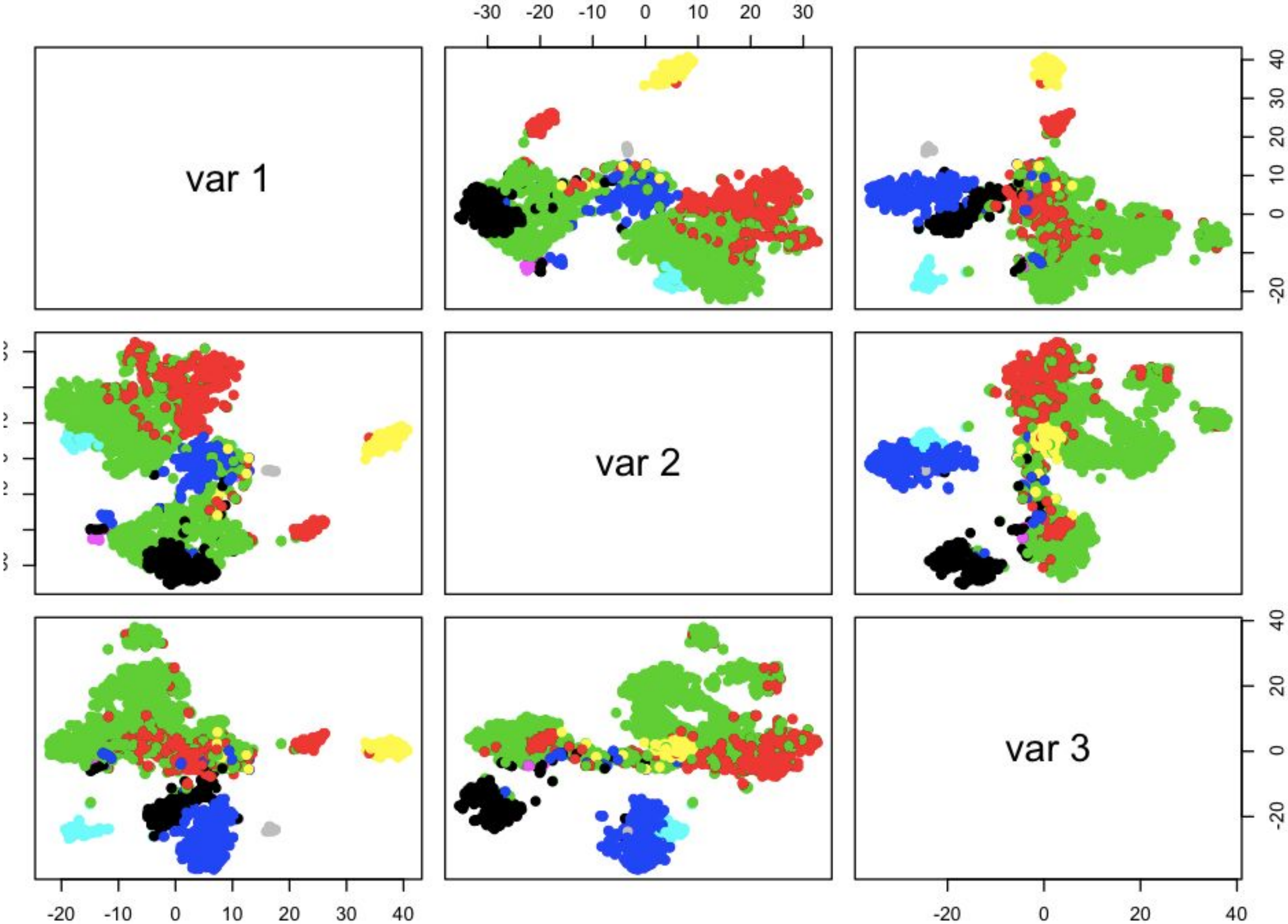
Left: default Rtsne on the 'conquer' quantifications for 739 genes; GEO notations
Right: as published in Darmanis 2017
Layouts different but concordant in various ways (three groups of 'neoplastic' [green], 'myeloid/immune' is 'largest' group, etc.)





Allow t-SNE to reduce to 3 dimensions instead of 2

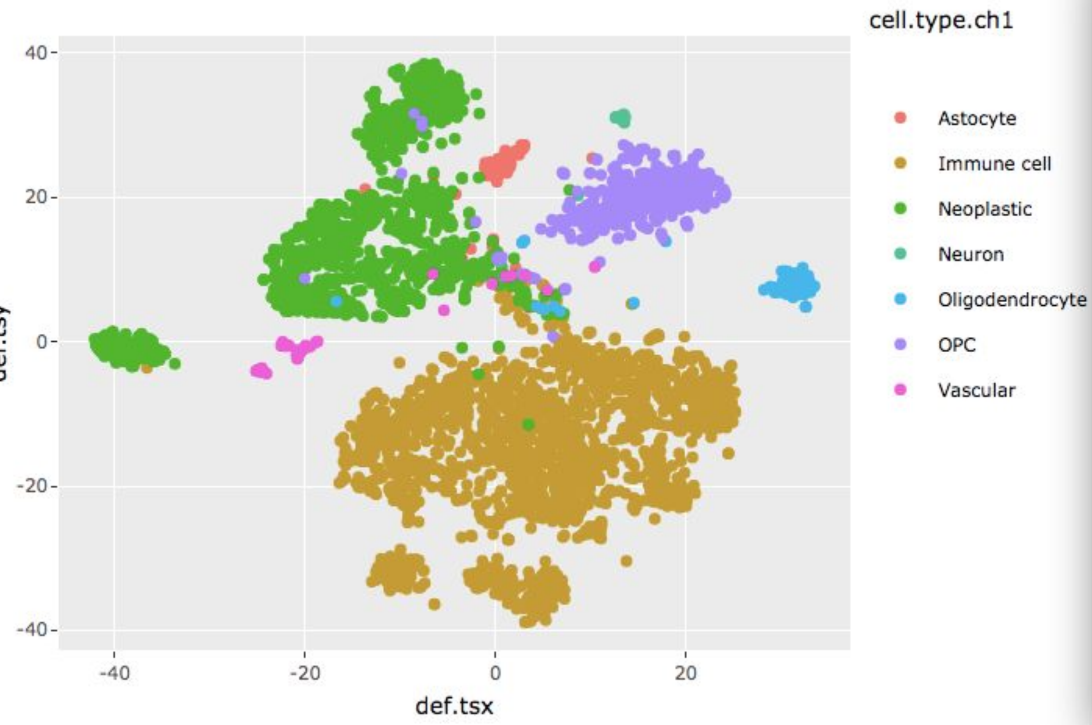
"topology" and "cluster relations" somewhat qualitatively different from the 2-d display

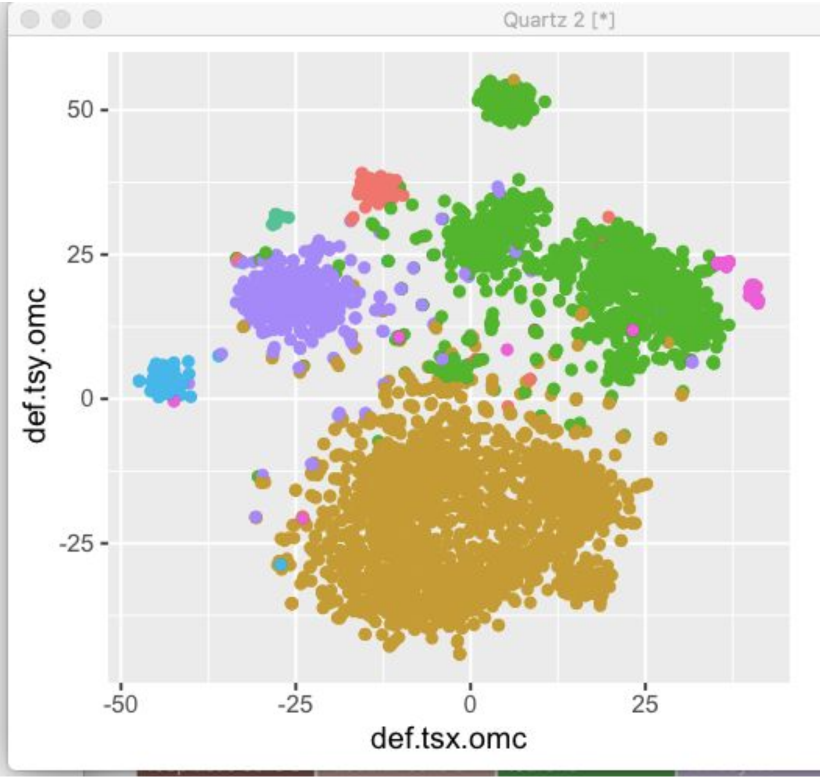


A dynamic graphic addressing this concern

try vjccc::spin_tsne()

Back to 2D: Left: Rtsne default euclidean distance Right: Use 1-cor distance and is_distance=TRUE





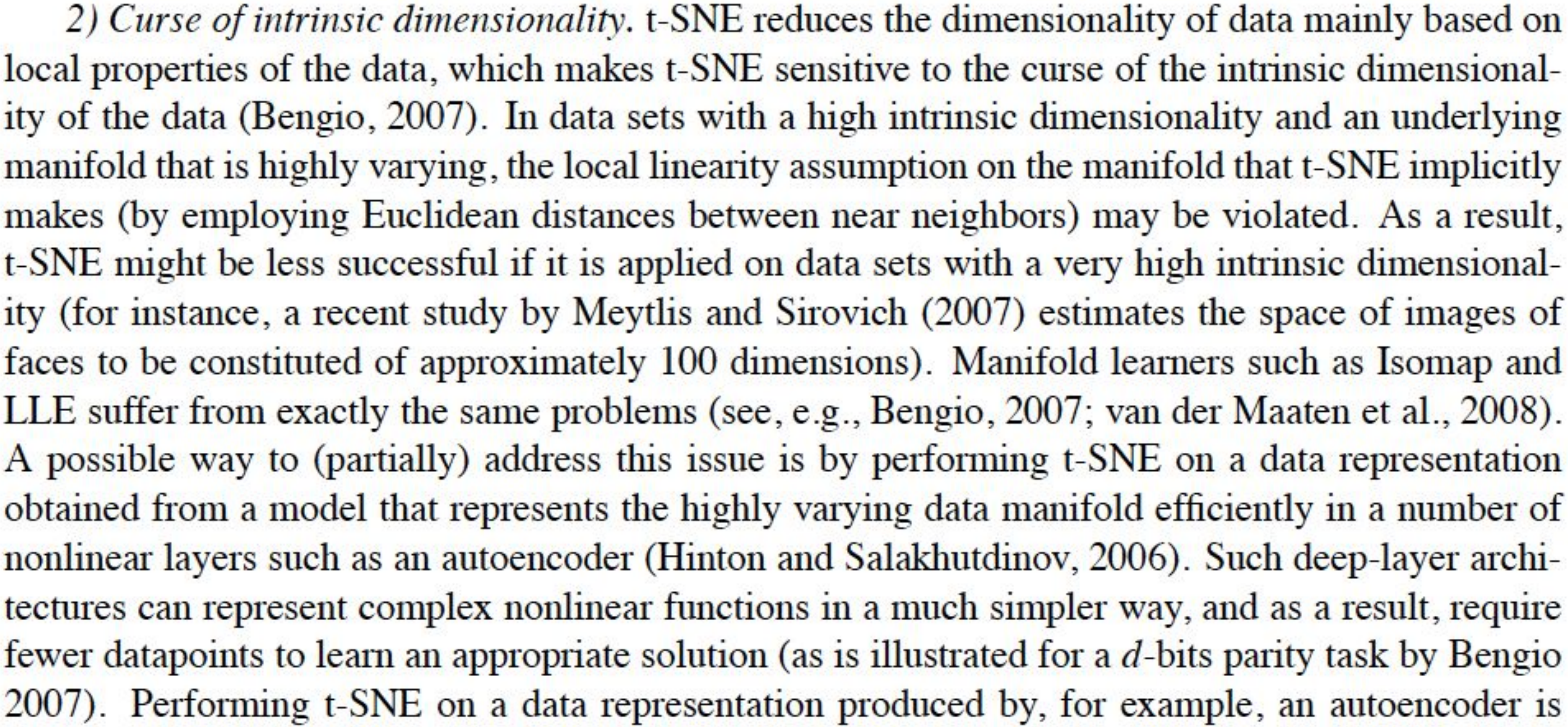
Caveats

- A trio of researchers from Google wrote a 'distill' paper
- https://distill.pub/2016/misread-tsne/
- Let's scroll quickly through it
- An issue for sensitivity analysis -- exploring various parameter settings -- is that the algorithm can take time to converge, you don't know when it has converged, and the hyperparameter space is potentially large

Can t-SNE have any value **at all** in complex biological systems?

- Depends on the objective
- "Proof of concept":
    - Winner of Merck Viz challenge 2011 (kaggle docs taken down?)
    - MNIST -- "digit separation"
    - flow cytometry *identifications* recapitulated with RNA-seq
- Essential assumption
    - a low dimensional structure exists and can be found with the iterative computation of 'similarities' leading to a minimum in the t-SNE objective function -- **global minimum need not exist**
    - **the tuning parameters are properly selected**

from the original paper by van der Maaten and Hinton:



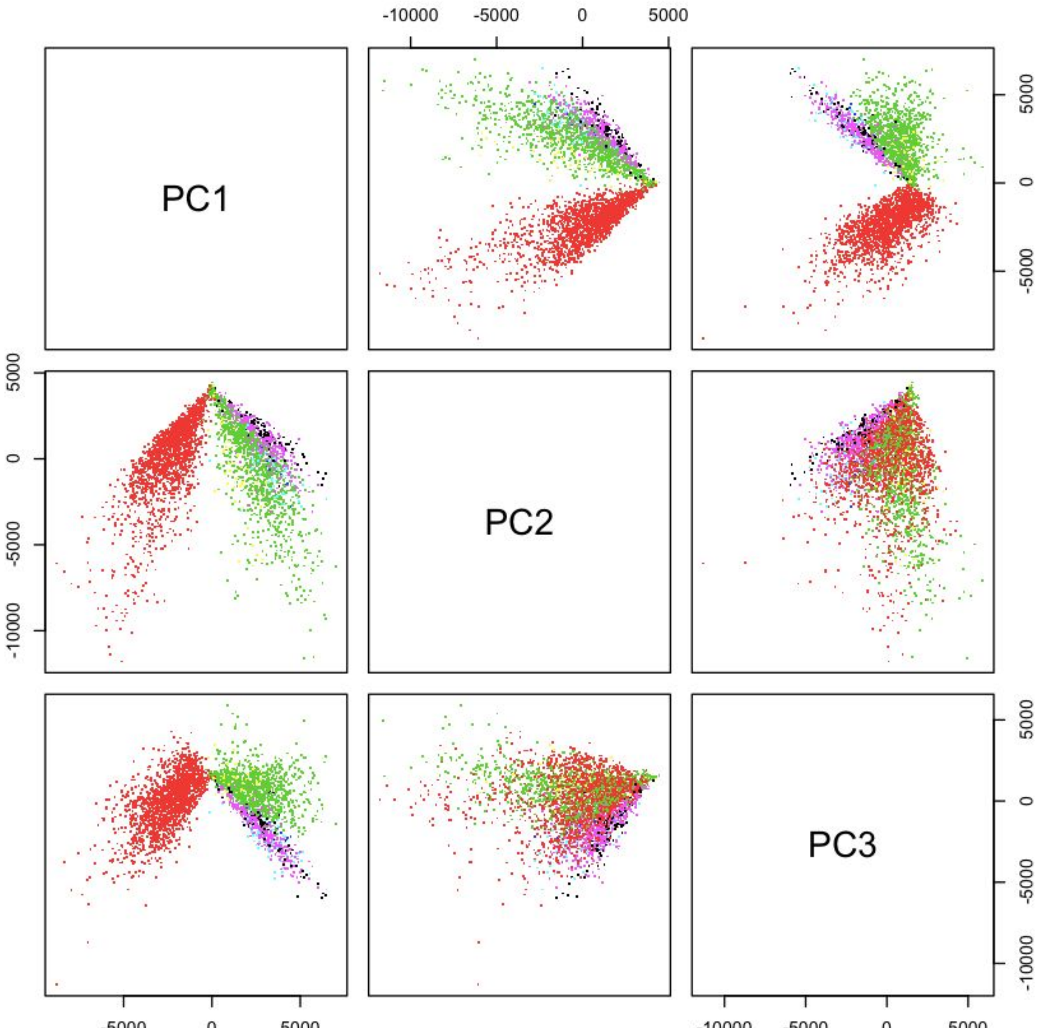
2) Curse of intrinsic dimensionality. t-SNE reduces the dimensionality of data mainly based on local properties of the data, which makes t-SNE sensitive to the curse of the intrinsic dimensionality of the data (Bengio, 2007). In data sets with a high intrinsic dimensionality and an underlying manifold that is highly varying, the local linearity assumption on the manifold that t-SNE implicitly makes (by employing Euclidean distances between near neighbors) may be violated. As a result, t-SNE might be less successful if it is applied on data sets with a very high intrinsic dimensionality (for instance, a recent study by Meytlis and Sirovich (2007) estimates the space of images of faces to be constituted of approximately 100 dimensions). Manifold learners such as Isomap and LLE suffer from exactly the same problems (see, e.g., Bengio, 2007; van der Maaten et al., 2008). A possible way to (partially) address this issue is by performing t-SNE on a data representation obtained from a model that represents the highly varying data manifold efficiently in a number of nonlinear layers such as an autoencoder (Hinton and Salakhutdinov, 2006). Such deep-layer architectures can represent complex nonlinear functions in a much simpler way, and as a result, require fewer datapoints to learn an appropriate solution (as is illustrated for a *d*-bits parity task by Bengio 2007). Performing t-SNE on a data representation produced by, for example, an autoencoder is

How does PCA reduce dimensionality?

```
pca1  =
prcomp(t(assay(se)))
```

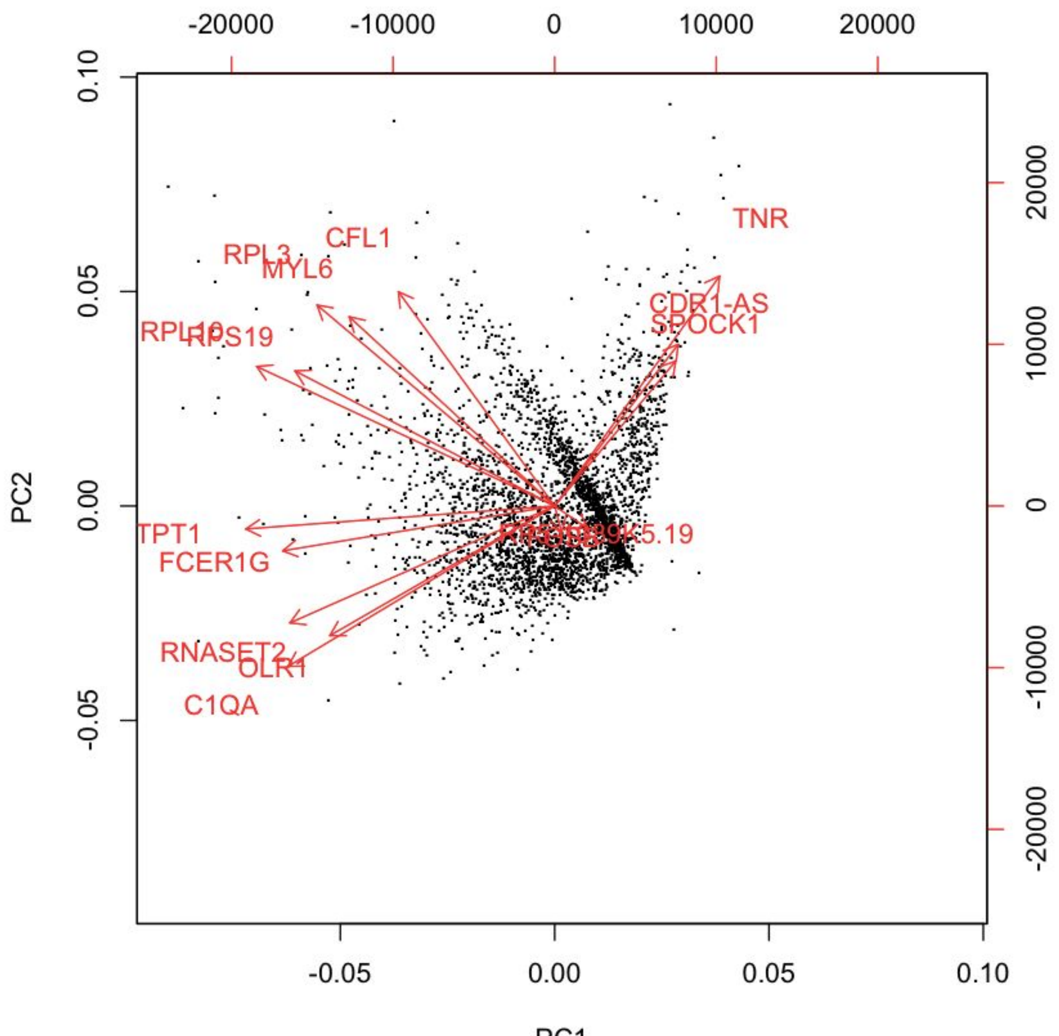
Here se is the 739 gene subset of Darmanis CONQUER

pairs(pca1$x[,1:3], …) [color is declared cell type]



biplots are useful but manual intervention often needed

Here I used elements of pca1$rotation to identify genes with relatively large 'loadings' and recomputed PCA with this subset to get a simpler biplot



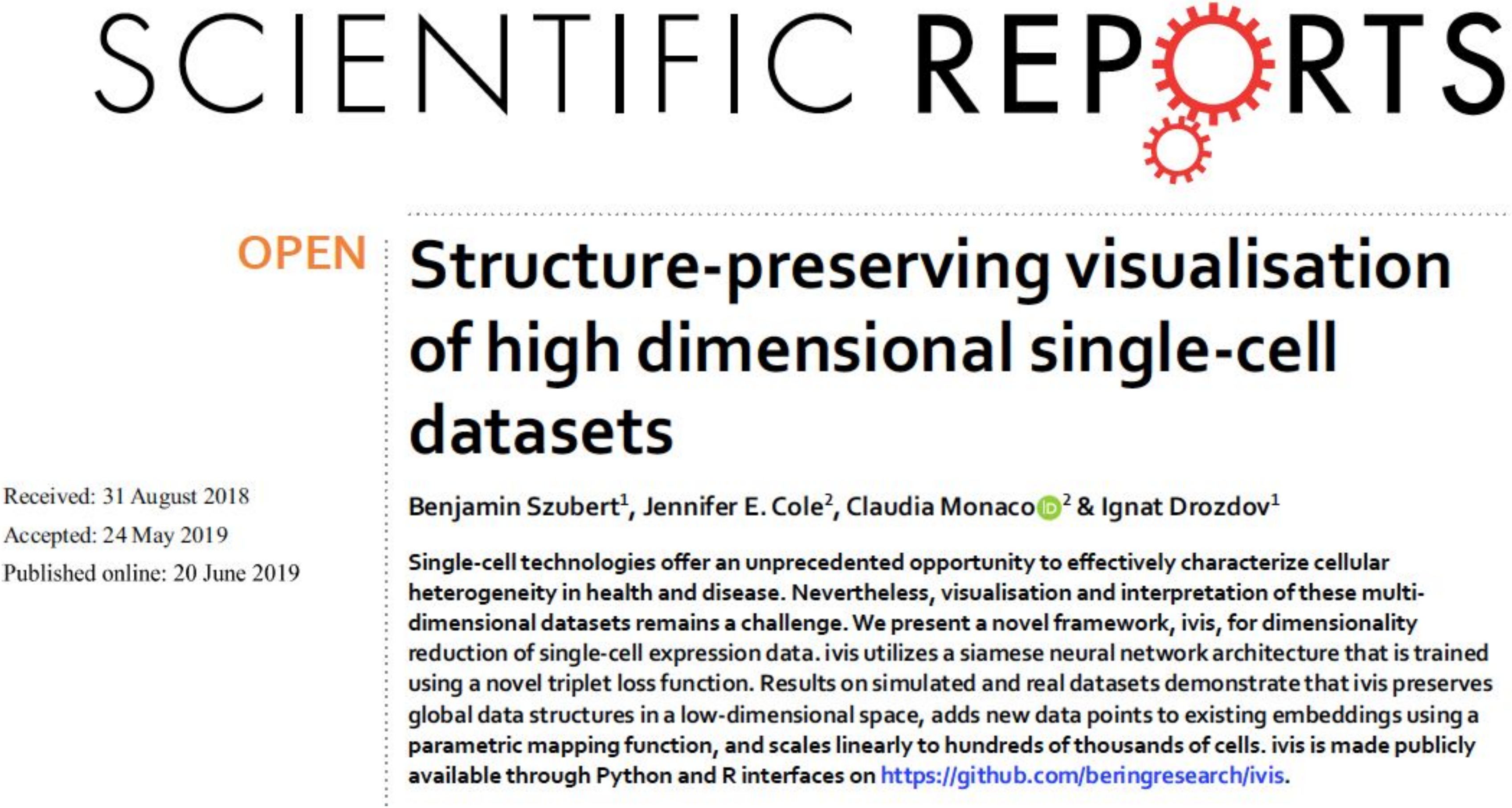
In what sense is PCA "feature engineering"?

For column-centered data matrix X, we can derive PCs using the singular value decomposition

$$X_{nxp} = UDV^t$$

in which columns of U are the PCs and columns of (orthonormal) V are loadings; D is diagonal with elements measuring variances of the corresponding PCs. Elements of columns of U are new features formed by linear combination of columns of X: $XVD^{-1} = U$ … and we use magnitudes of elements of D to determine how many PCs are "needed" to approximate variation in X

Comparing approaches to dimension reduction



SCIENTIFIC REPORTS

OPEN

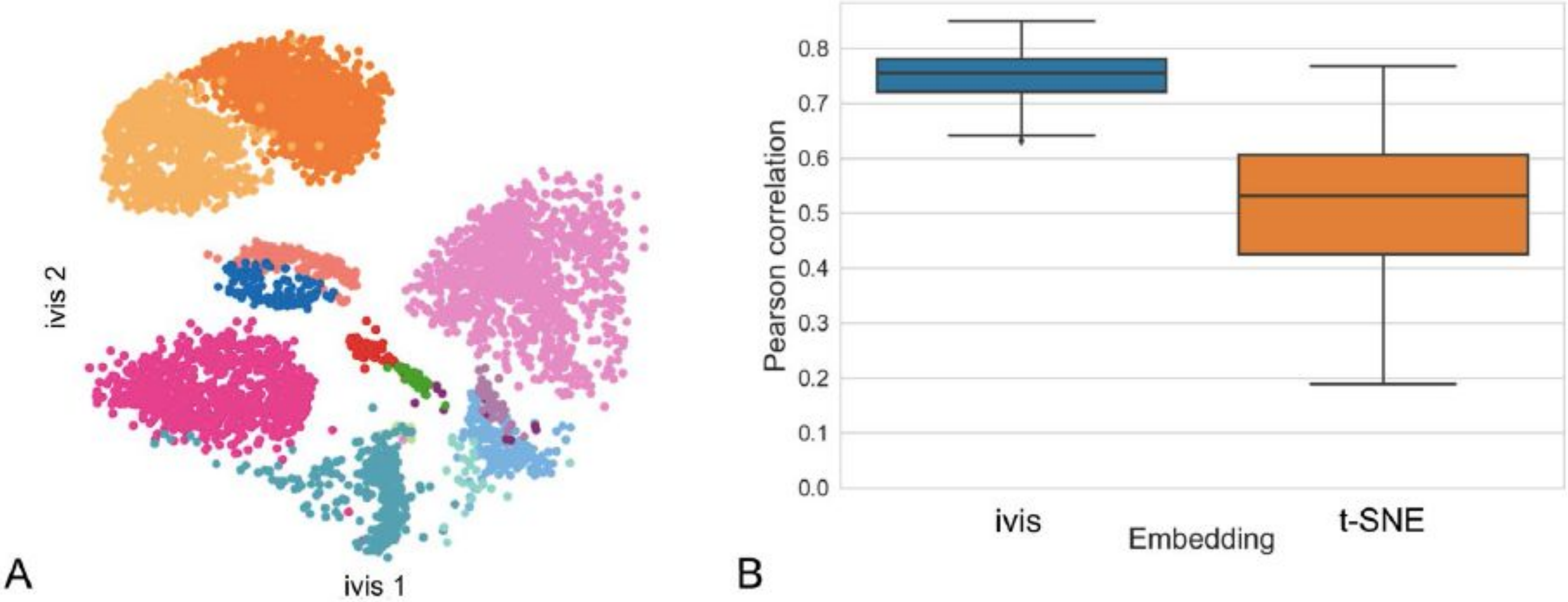
Structure-preserving visualisation of high dimensional single-cell datasets

Received: 31 August 2018
Accepted: 24 May 2019
Published online: 20 June 2019

Benjamin Szubert[1], Jennifer E. Cole[2], Claudia Monaco[2] & Ignat Drozdov[1]

Single-cell technologies offer an unprecedented opportunity to effectively characterize cellular heterogeneity in health and disease. Nevertheless, visualisation and interpretation of these multi-dimensional datasets remains a challenge. We present a novel framework, ivis, for dimensionality reduction of single-cell expression data. ivis utilizes a siamese neural network architecture that is trained using a novel triplet loss function. Results on simulated and real datasets demonstrate that ivis preserves global data structures in a low-dimensional space, adds new data points to existing embeddings using a parametric mapping function, and scales linearly to hundreds of thousands of cells. ivis is made publicly available through Python and R interfaces on https://github.com/beringresearch/ivis.

ivis compared to t-SNE: correlating distances between asserted cluster centers and centers given by manually gating in cyTOF



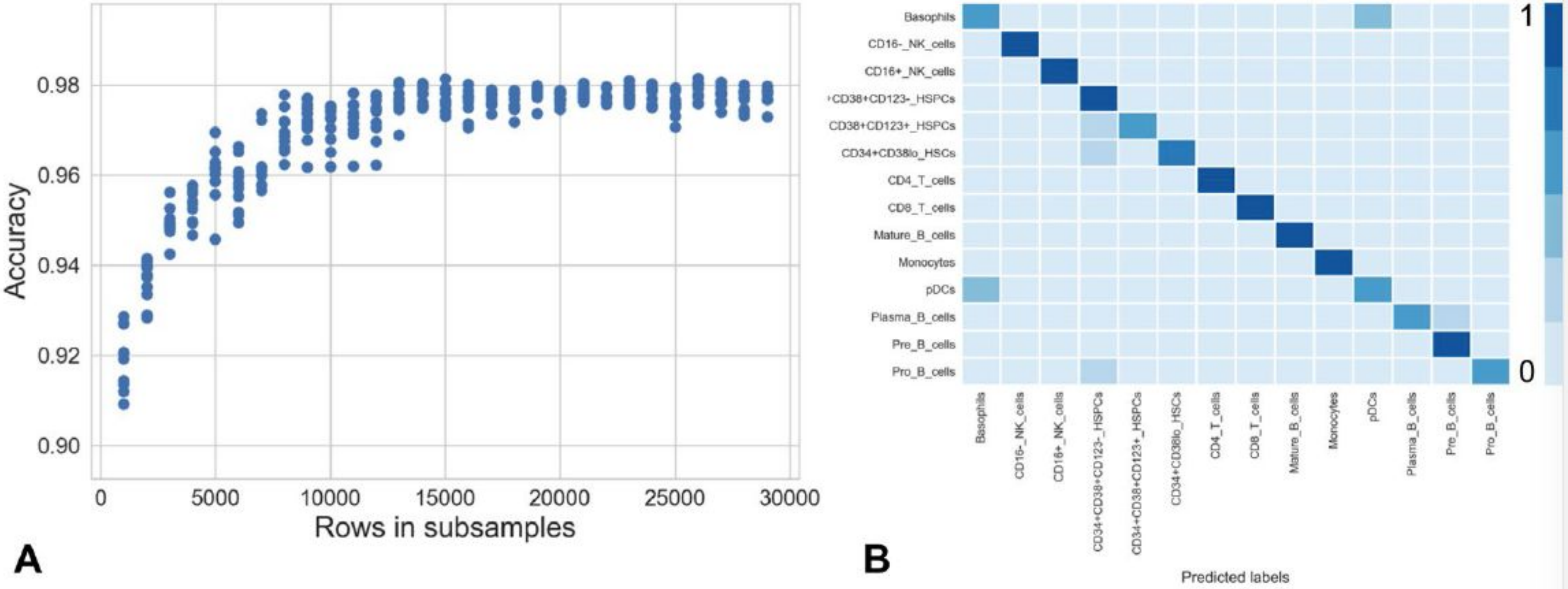
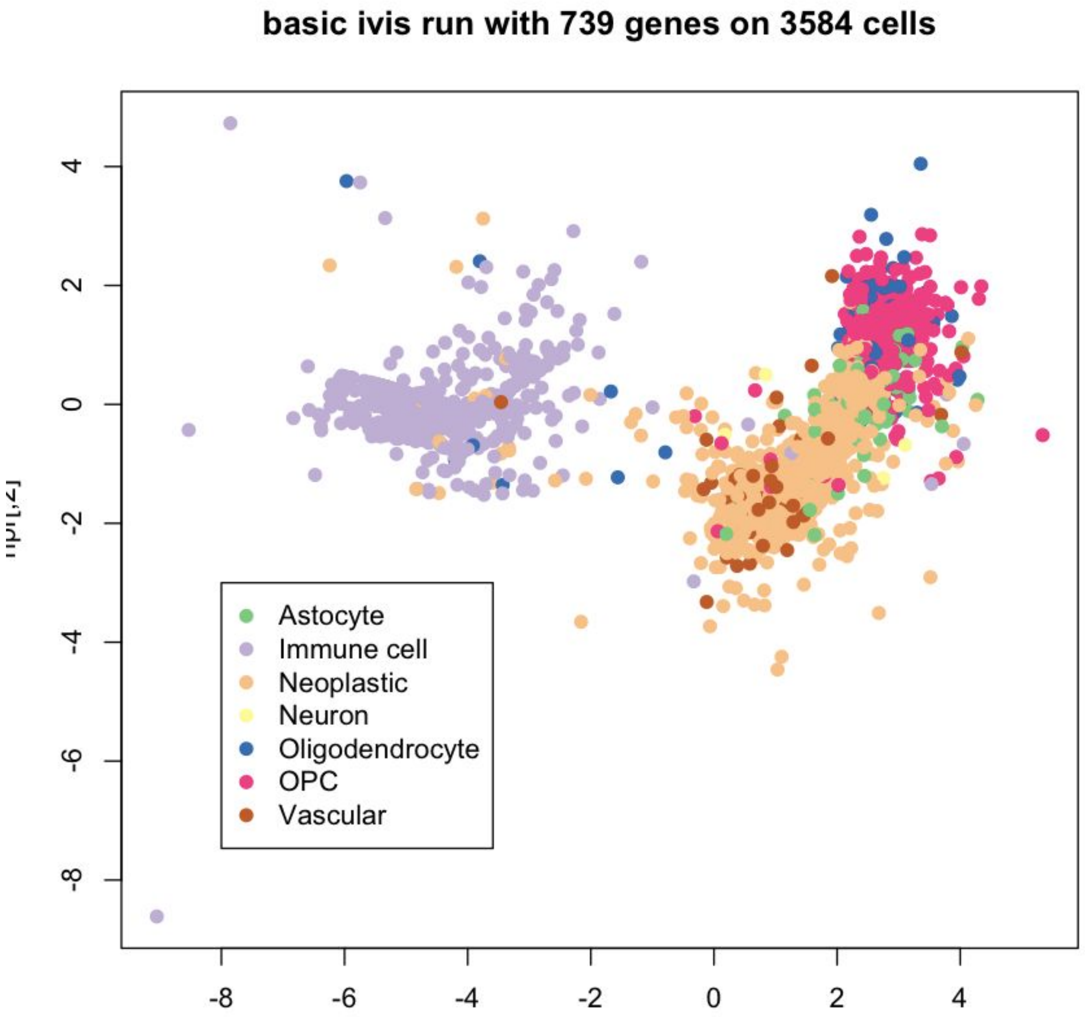


Figure 6. Random forest classifier performance on ivis embeddings inferred from independent subsets of healthy human BMMC data. (**A**) Scatterplot depicting accuracy of a random forest classifier when trained on embedded subsets of varying size. The experiments for each subset size were repeated ten times. (**B**) Confusion matrix for a single random forest classifier trained on a subset of 10,000 embedded data-points and validated o

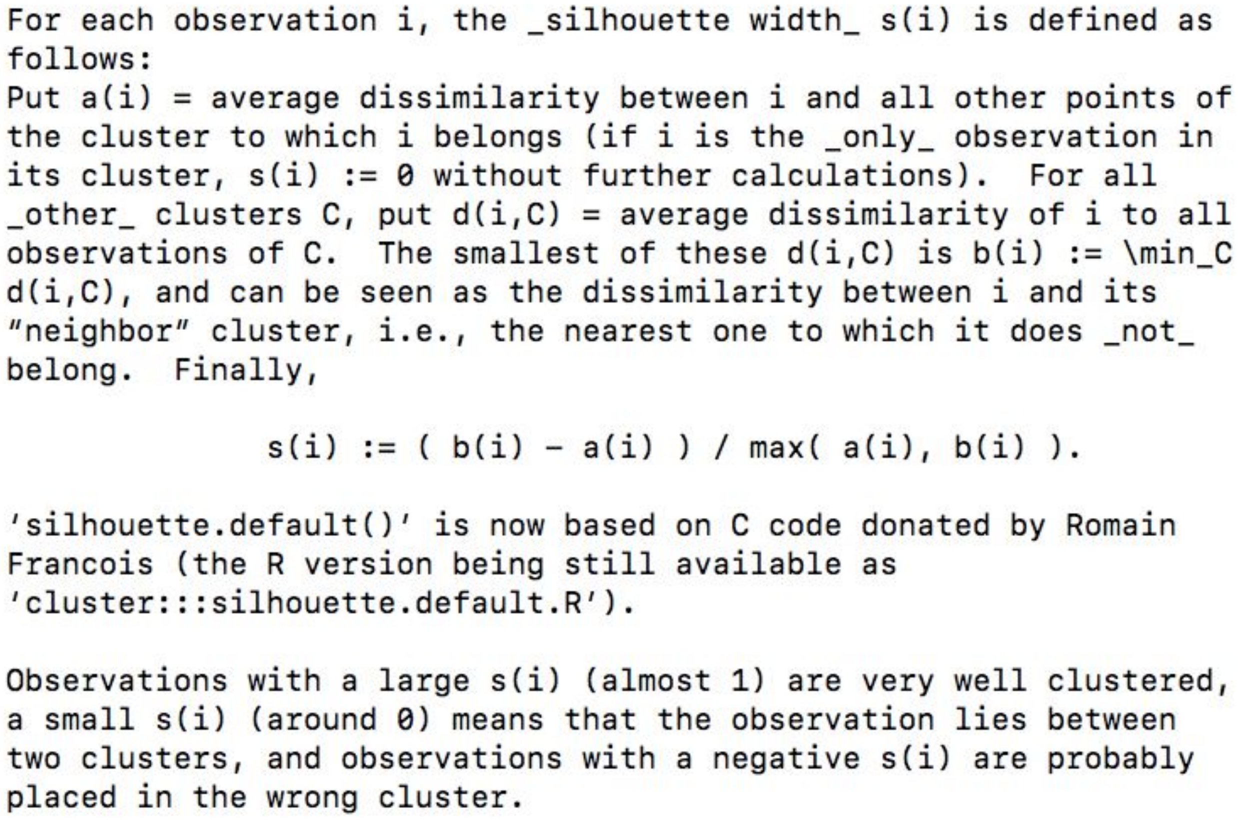
If ivis is actually superior to t-SNE at accurately exhibiting lower-dimensional structures from high-dimensional data, then it might be concluded from this display that the tumor cells acquired in the Darmanis study divide into only two transcriptomically-defined groups

However tuning and selection of target dimensionality demand additional attention



Use and figures of merit for cluster analysis

- Basic measure of cluster coherence: silhouette

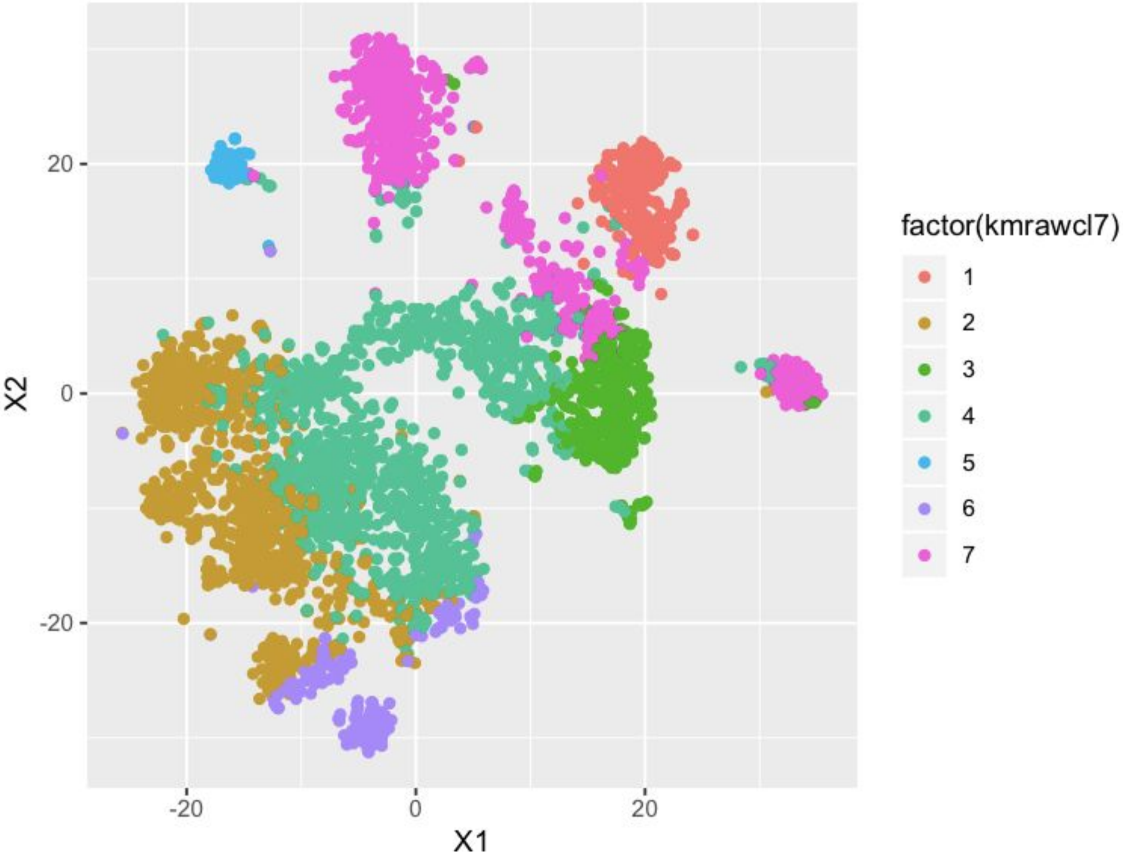


```
For each observation i, the _silhouette width_ s(i) is defined as
follows:
Put a(i) = average dissimilarity between i and all other points of
the cluster to which i belongs (if i is the _only_ observation in
its cluster, s(i) := 0 without further calculations).  For all
_other_ clusters C, put d(i,C) = average dissimilarity of i to all
observations of C.  The smallest of these d(i,C) is b(i) := \min_C
d(i,C), and can be seen as the dissimilarity between i and its
"neighbor" cluster, i.e., the nearest one to which it does _not_
belong.  Finally,

              s(i) := ( b(i) - a(i) ) / max( a(i), b(i) ).

'silhouette.default()' is now based on C code donated by Romain
Francois (the R version being still available as
'cluster:::silhouette.default.R').

Observations with a large s(i) (almost 1) are very well clustered,
a small s(i) (around 0) means that the observation lies between
two clusters, and observations with a negative s(i) are probably
placed in the wrong cluster.
```

Order of dimension reduction and clustering

Typical procedure is to use clustering after dimension reduction?

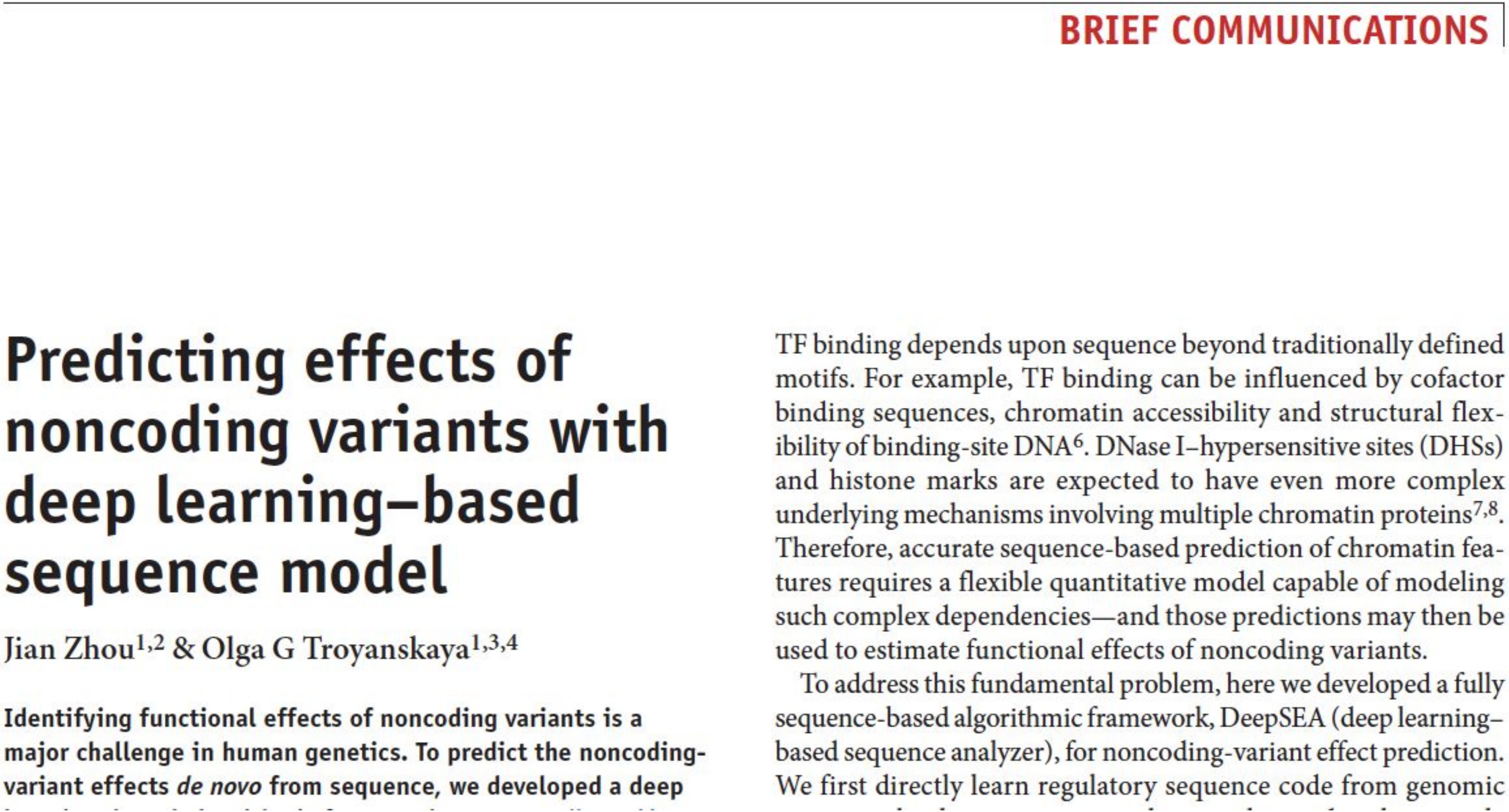
If we already reduce the feature set to hundreds of genes **before** dimension reduction, we might cluster with those … example to right



"Classification" -- lots of material ready to hand for self-study

https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf

instead we take a quick look at reusable deep learning!



Predicting effects of noncoding variants with deep learning–based sequence model

Jian Zhou[1,2] & Olga G Troyanskaya[1,3,4]

Identifying functional effects of noncoding variants is a major challenge in human genetics. To predict the noncoding-variant effects *de novo* from sequence, we developed a deep

TF binding depends upon sequence beyond traditionally defined motifs. For example, TF binding can be influenced by cofactor binding sequences, chromatin accessibility and structural flexibility of binding-site DNA[6]. DNase I–hypersensitive sites (DHSs) and histone marks are expected to have even more complex underlying mechanisms involving multiple chromatin proteins[7,8]. Therefore, accurate sequence-based prediction of chromatin features requires a flexible quantitative model capable of modeling such complex dependencies—and those predictions may then be used to estimate functional effects of noncoding variants.

To address this fundamental problem, here we developed a fully sequence-based algorithmic framework, DeepSEA (deep learning–based sequence analyzer), for noncoding-variant effect prediction. We first directly learn regulatory sequence code from genomic

Resource 1: a large number of epigenomic reference resources

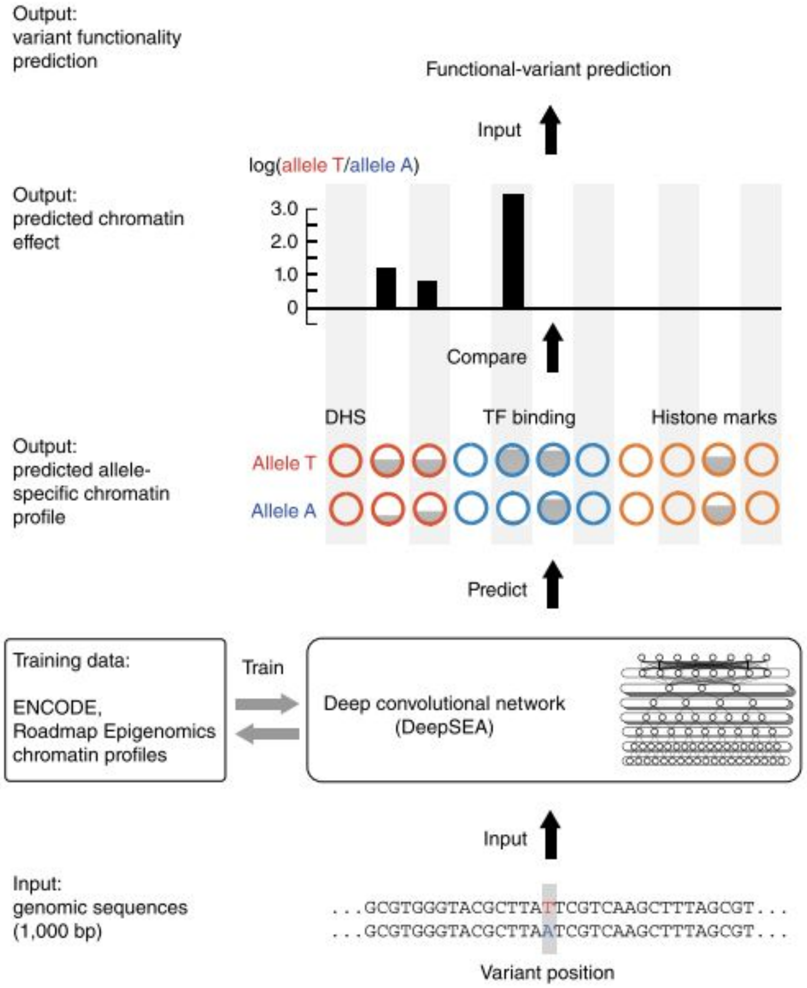
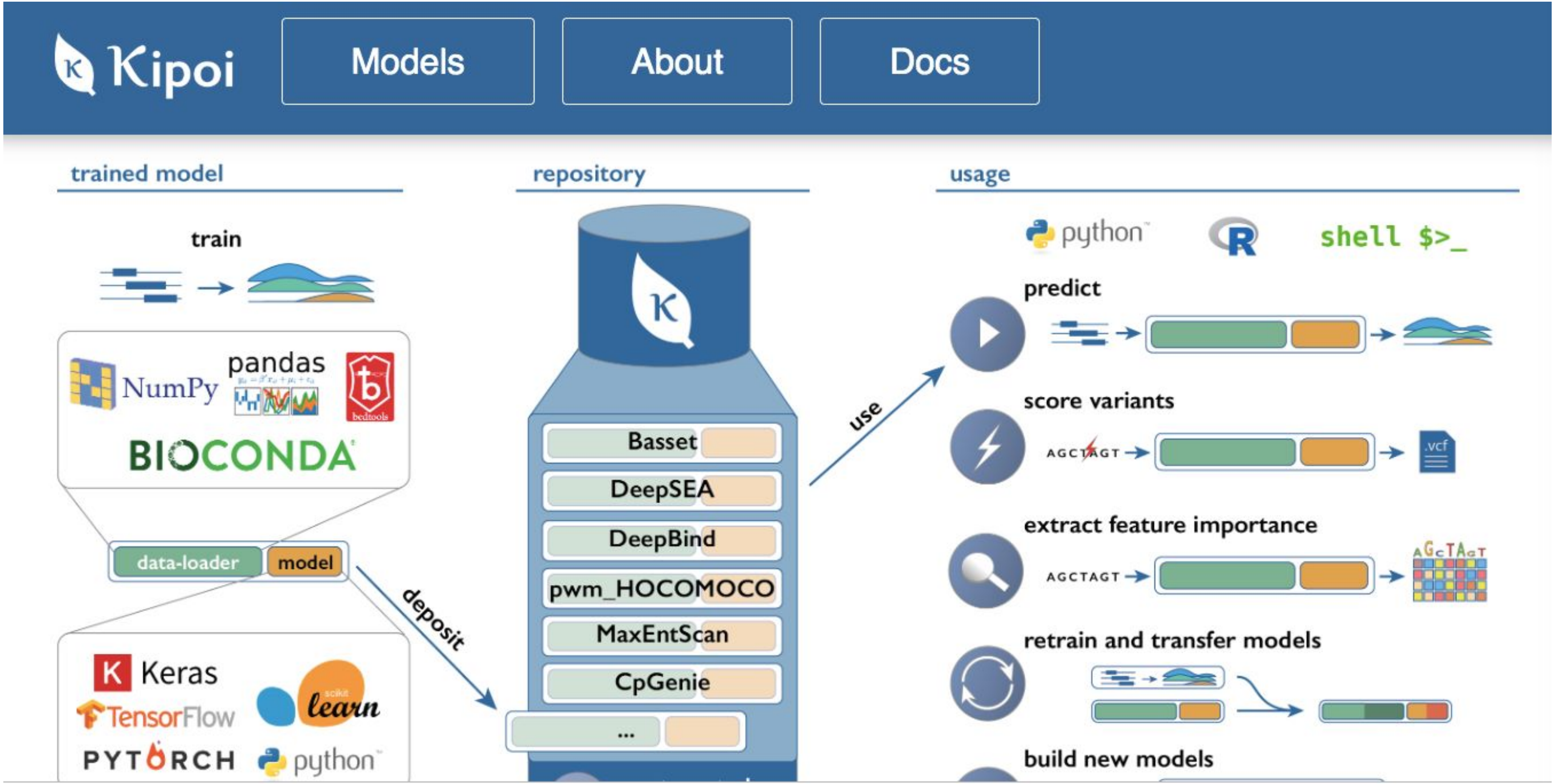
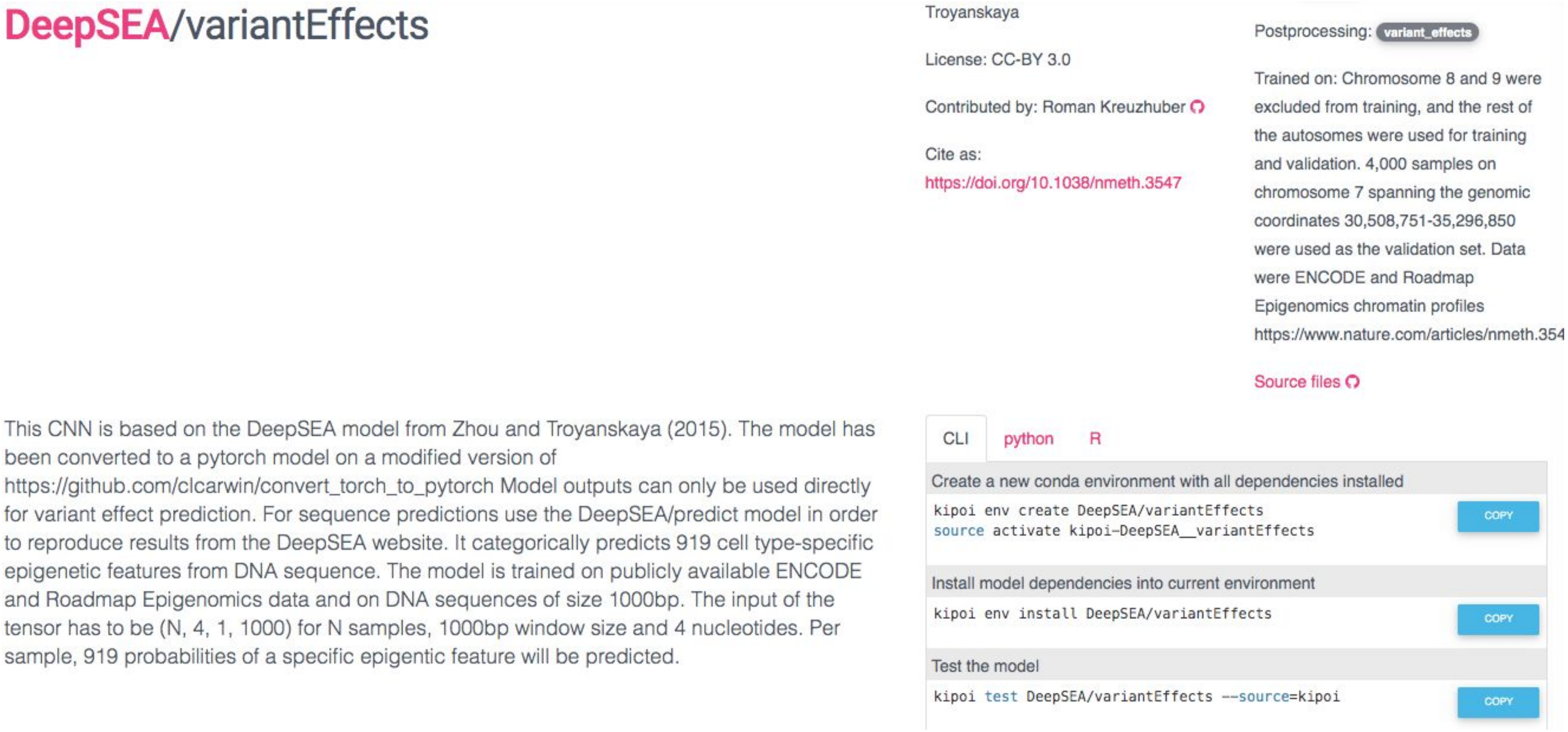


Figure 1 | Schematic overview of the DeepSEA pipeline, a strategy for predicting chromatin effects of noncoding variants.

Resource 2: net architecture and coefficient values


Kipoi
Models
About
Docs
trained model
train
NumPy
pandas
bedtools
BIOCONDA
data-loader
model
Keras
TensorFlow
scikit learn
PYTORCH
python
deposit
repository
Basset
DeepSEA
DeepBind
pwm_HOCOMOCO
MaxEntScan
CpGenie
...
use
usage
python
R
shell $>_
predict
score variants
AGCTAGT
.vcf
extract feature importance
AGCTAGT
retrain and transfer models
build new models



DeepSEA/variantEffects

Troyanskaya

License: CC-BY 3.0

Contributed by: Roman Kreuzhuber

Cite as:
https://doi.org/10.1038/nmeth.3547

Postprocessing: variant_effects

Trained on: Chromosome 8 and 9 were excluded from training, and the rest of the autosomes were used for training and validation. 4,000 samples on chromosome 7 spanning the genomic coordinates 30,508,751-35,296,850 were used as the validation set. Data were ENCODE and Roadmap Epigenomics chromatin profiles https://www.nature.com/articles/nmeth.354

Source files

This CNN is based on the DeepSEA model from Zhou and Troyanskaya (2015). The model has been converted to a pytorch model on a modified version of https://github.com/clcarwin/convert_torch_to_pytorch Model outputs can only be used directly for variant effect prediction. For sequence predictions use the DeepSEA/predict model in order to reproduce results from the DeepSEA website. It categorically predicts 919 cell type-specific epigenetic features from DNA sequence. The model is trained on publicly available ENCODE and Roadmap Epigenomics data and on DNA sequences of size 1000bp. The input of the tensor has to be (N, 4, 1, 1000) for N samples, 1000bp window size and 4 nucleotides. Per sample, 919 probabilities of a specific epigentic feature will be predicted.

Create a new conda environment with all dependencies installed

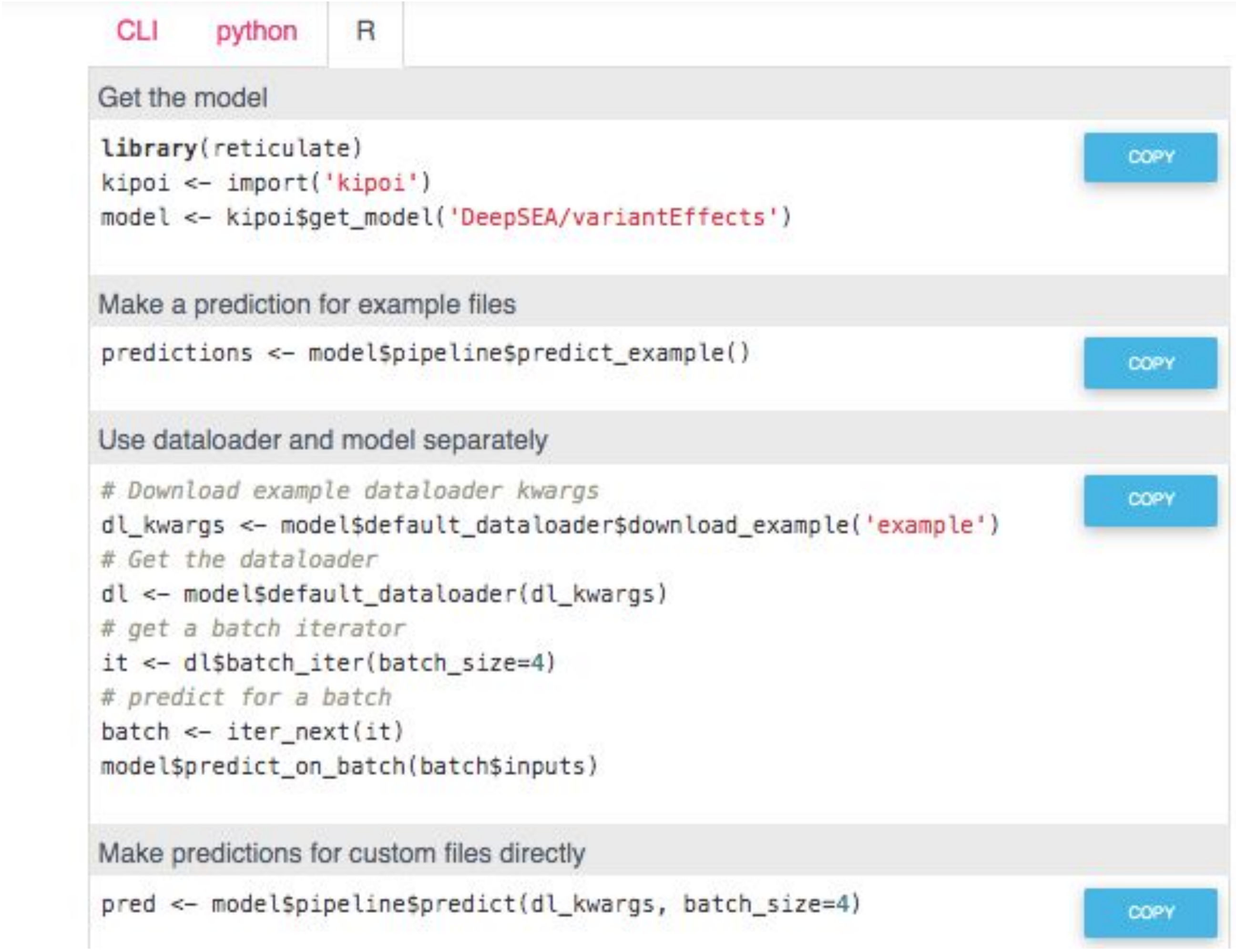
```
kipoi env create DeepSEA/variantEffects
source activate kipoi-DeepSEA__variantEffects
```

Install model dependencies into current environment

```
kipoi env install DeepSEA/variantEffects
```

Test the model

```
kipoi test DeepSEA/variantEffects --source=kipoi
```



Get the model

```
library(reticulate)
kipoi <- import('kipoi')
model <- kipoi$get_model('DeepSEA/variantEffects')
```

Make a prediction for example files

```
predictions <- model$pipeline$predict_example()
```

Use dataloader and model separately

```
# Download example dataloader kwargs
dl_kwargs <- model$default_dataloader$download_example('example')
# Get the dataloader
dl <- model$default_dataloader(dl_kwargs)
# get a batch iterator
it <- dl$batch_iter(batch_size=4)
# predict for a batch
batch <- iter_next(it)
model$predict_on_batch(batch$inputs)
```

Make predictions for custom files directly

```
pred <- model$pipeline$predict(dl_kwargs, batch_size=4)
```

- Couldn't be easier to get acquainted with capabilities of 'deep learning'
- downloads to set up models and infrastructure can take some time
- inter-language interface can be opaque

Upshots

- Various resources for model definition and fitting stored in zenodo and AWS S3
- interfaces to tensorflow etc. set up for use in CLI, python, R
- understanding how to
  - deploy against new data
  - update model with new reference data
  - contribute de novo models to this reusability framework
  - exercises!
- I have observed that some models of interest don't work as advertised, but deepSEA example did work. Could be a continuous integration issue

Summary

- Case study: single-cell RNA-seq in glioblastoma
  - use CONQUER, try Rtsne, smoothing expression, etc.
- Distances and the curse of dimensionality
  - many ad hoc approaches, check sensitivity to assumptions
- Dimension reduction and feature engineering
  - t-SNE, ivis, PCA -- biplots are interpretable
  - framework for comparing feature engineering methods is urgently needed but very hard

- Options and figures of merit in cluster analysis
- Concepts of supervised learning -- use ESL_II
- kipoi.org: an archive of trained models