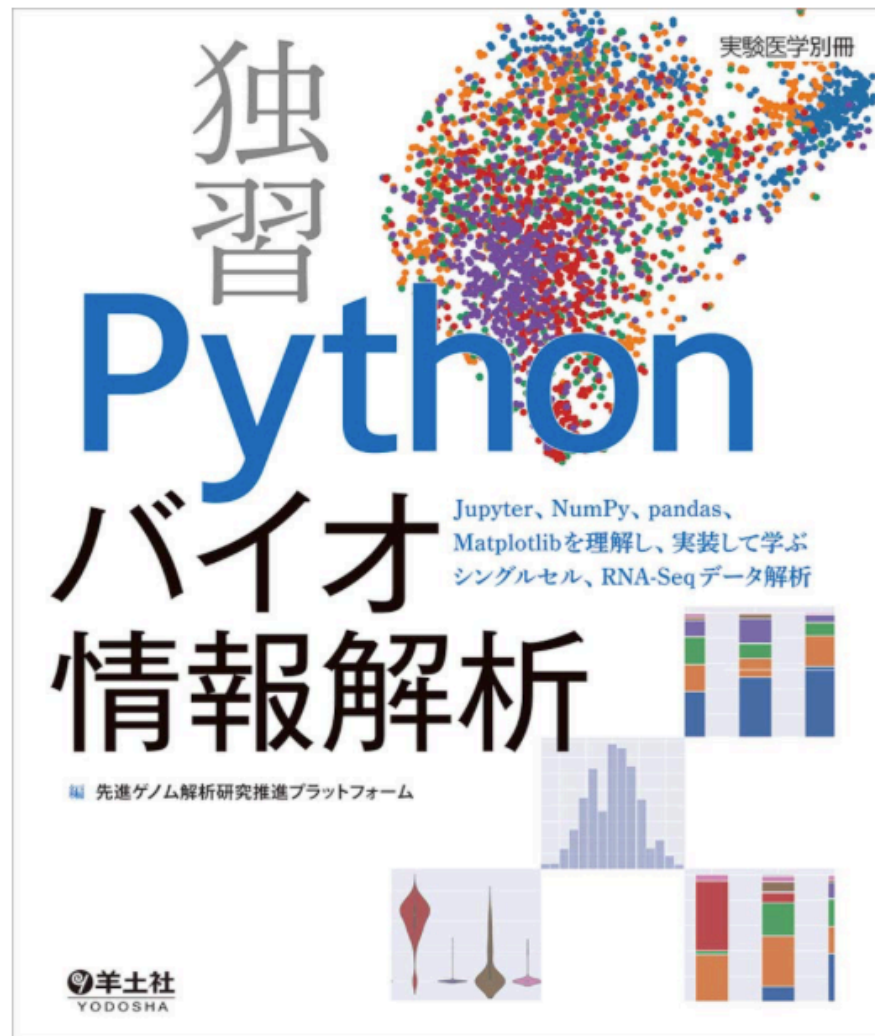


Pythonを用いたシングルセルRNA-seq解析（基礎）

2024/02/14
遺伝研 望月孝子

謝辞

本講習の内容は、2019年度、2023年度の先進ゲノム支援の情報解析講習会 遺伝研・東光一先生の資料、ご発表の内容をベースにさせて頂いております。



独習 Python バイオ情報解析の内容でも勉強させて頂きました。

scRNA-seq の解析のステップにおいて、なぜその処理が必要なのか？そのアルゴリズムはどうなっているのか？を説明して下さいで大変参考になりました。

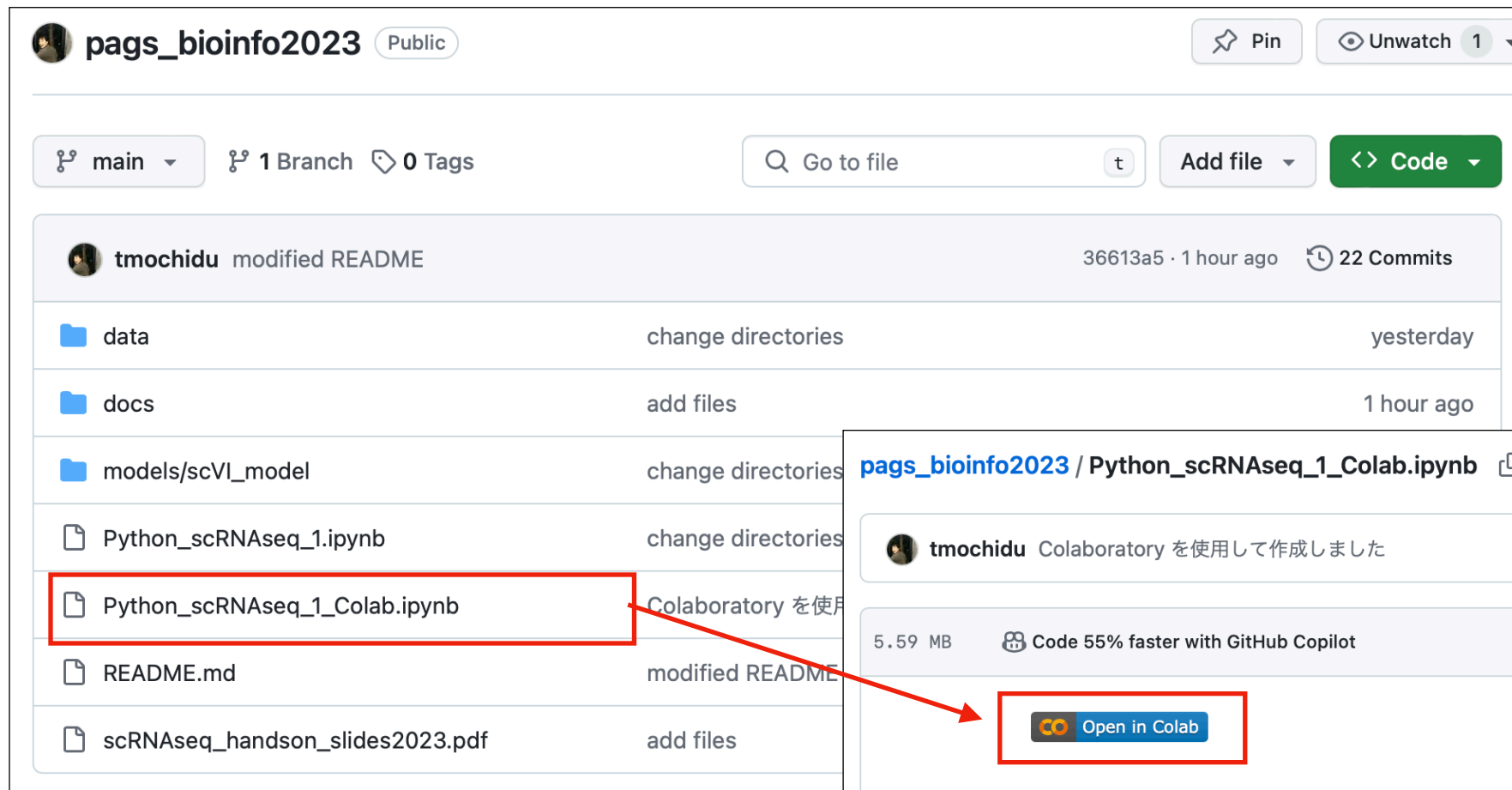
東先生、ありがとうございました。

ご注意ください。

- 「ベストプラクティス」というわけではありません。発展的なツールの紹介もあって、中には有効性がまだじゅうぶんに検証されていないものもある。

- M1チップのMacなど、必要パッケージがインストールできなかった場合は Docker か、Google Colabのバージョンで実行。

https://github.com/tmochidu/pags_bioinfo2023



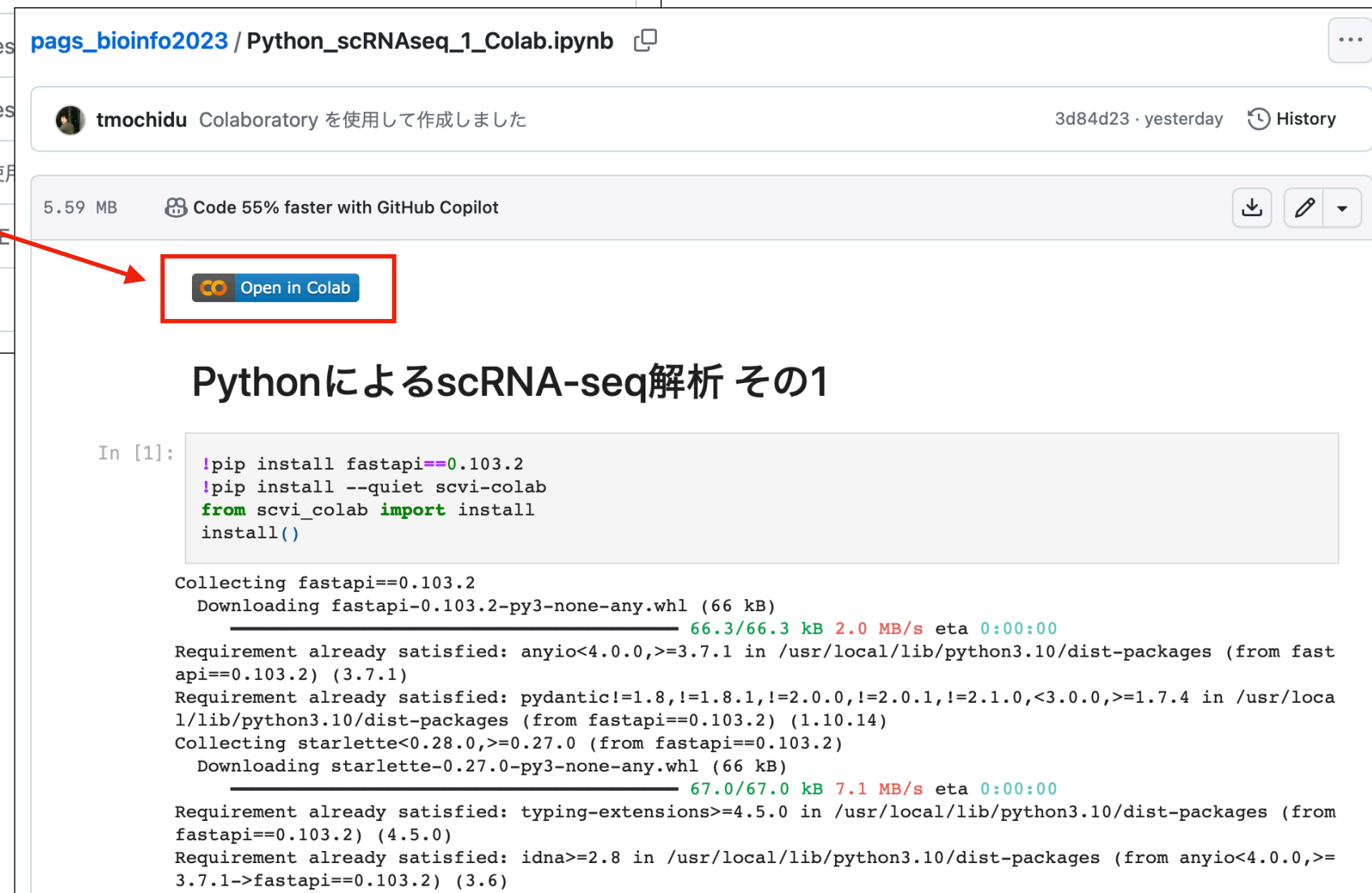
Repository: **pags_bioinfo2023** (Public)

main 1 Branch 0 Tags

Go to file t Add file <> Code

tmochidu modified README 36613a5 · 1 hour ago 22 Commits

- data change directories yesterday
- docs add files 1 hour ago
- models/scVI_model change directories
- Python_scRNAseq_1.ipynb change directories
- Python_scRNAseq_1_Colab.ipynb** Colaboratory を使用
- README.md modified README
- scRNAseq_handson_slides2023.pdf add files



pags_bioinfo2023 / Python_scRNAseq_1_Colab.ipynb

tmochidu Colaboratory を使用して作成しました 3d84d23 · yesterday History

5.59 MB Code 55% faster with GitHub Copilot

Open in Colab

PythonによるscRNA-seq解析 その1

In [1]:

```
!pip install fastapi==0.103.2
!pip install --quiet scvi-colab
from scvi_colab import install
install()
```

Collecting fastapi==0.103.2
Downloading fastapi-0.103.2-py3-none-any.whl (66 kB)
66.3/66.3 kB 2.0 MB/s eta 0:00:00

Requirement already satisfied: anyio<4.0.0,>=3.7.1 in /usr/local/lib/python3.10/dist-packages (from fastapi==0.103.2) (3.7.1)

Requirement already satisfied: pydantic!=1.8,!<1.8.1,!<2.0.0,!<2.0.1,!<2.1.0,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from fastapi==0.103.2) (1.10.14)

Collecting starlette<0.28.0,>=0.27.0 (from fastapi==0.103.2)

Downloading starlette-0.27.0-py3-none-any.whl (66 kB)
67.0/67.0 kB 7.1 MB/s eta 0:00:00

Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.10/dist-packages (from fastapi==0.103.2) (4.5.0)

Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<4.0.0,>=3.7.1->fastapi==0.103.2) (3.6)

Google Colab で実行する方

手順2

The screenshot shows the Google Colab interface for a notebook titled 'Python_scRNAseq_1_Colab.ipynb'. A warning dialog box is displayed in the center, stating: '警告: このノートブックは Google が作成したものではありません。' (Warning: This notebook was not created by Google). Below this, it says: 'このノートブックは [GitHub](#) から読み込まれています。Google に保存されているデータへのアクセスが求められたり、他のセッションからデータや認証情報が読み取られたりする場合があります。このノートブックを実行する前にソースコードをご確認ください。' (This notebook is loaded from [GitHub](#). There may be requests for access to data saved in Google or for data or authentication information to be retrieved from other sessions. Please check the source code before running this notebook). At the bottom of the dialog are 'キャンセル' (Cancel) and 'このまま実行' (Run anyway) buttons. In the background, a code cell is visible with the following code:

```
!pip install fastapi==0.103.2
!pip install --quiet scvi-colab
from scvi_colab import install
install()
```

Google アカウントにログイン。
遺伝研アカウントでは実行制限さ
れているので個人アカウントで。

3つ目のこのセルまで実行

```
[ ] !pip install --upgrade pandas
```

3つ目のセル実行後、
メニュー「ランタイム」->「セッ
ションを再起動する」をクリックし
てください。

The screenshot shows the Google Colab interface with the 'Runtime' menu open. The menu options are: 'すべてのセルを実行' (Run all cells), 'より前のセルを実行' (Run previous cells), '現在のセルを実行' (Run current cell), '選択範囲を実行' (Run selection), '以降のセルを実行' (Run remaining cells), '実行を中断' (Interrupt execution), 'セッションを再起動する' (Restart session), 'セッションを再起動してすべて実行する' (Restart session and run all), 'ランタイムを接続解除して削除' (Disconnect and delete runtime), 'ランタイムのタイプを変更' (Change runtime type), 'セッションの管理' (Manage sessions), 'リソースを表示' (View resources), and 'ランタイムログの表示' (View runtime logs). The 'セッションを再起動する' option is highlighted with a red box. Below the menu, a red box contains the text: 'ここでランタイムの再起動！' (Restart runtime here!).

```
[1] !pip install --quiet scvi-colab
    from scvi_colab import install
    install()
```

```
INFO      scvi-colab: Installing scvi-tools.
INFO      scvi-colab: Install successful. Testing import.
/usr/local/lib/python3.10/dist-packages/scvi/_settings.py:63: UserWarning: Since
self.seed = seed
/usr/local/lib/python3.10/dist-packages/scvi/_settings.py:70: UserWarning: Setti
self.dl_pin_memory_gpu_training = (
```

```
[2] !git clone https://github.com/tmochidu/pags\_bioinfo2023.git
```

```
Cloning into 'pags_bioinfo2023'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 79 (delta 0), reused 0 (delta 0), pack-reused 76
Receiving objects: 100% (79/79), 107.07 MiB | 28.42 MiB/s, done.
Resolving deltas: 100% (15/15), done.
```

```
[3] %cd /content/pags\_bioinfo2023
```

```
/content/pags_bioinfo2023
```

その後、上記3つのセルを実行すれば、google colab を使わないで実習を行う方と同じところからスタートできます。



genome-sci / python_bioinfo_2023

Docker を利用する場合

```
# ソースコード取得
https://github.com/genome-sci/python_bioinfo_2023.git

# ディレクトリに移動
cd python_bioinfo_2023

# コンテナイメージをビルド（pags:2023という名称のイメージを作成します）
docker build -t pags:2023 .

# jupyter notebook 起動
docker run -it --rm -v $PWD:/python_bioinfo_2023 -p 8888:8888 pags:2023

# コンテナ内でコマンドを実行する
docker run -it --rm -v $PWD:/python_bioinfo_2023 -p 8888:8888 pags:2023 bash
```

Jupyter notebook を開くには起動後、ウェブブラウザから <http://localhost:8888> を開いてください。

Jupyter notebook で 「Python_scRNAseq_1.ipynb」 を起動してください。

ターミナルで以下のコマンドを実行してください。

```
$ conda activate pags2023
```

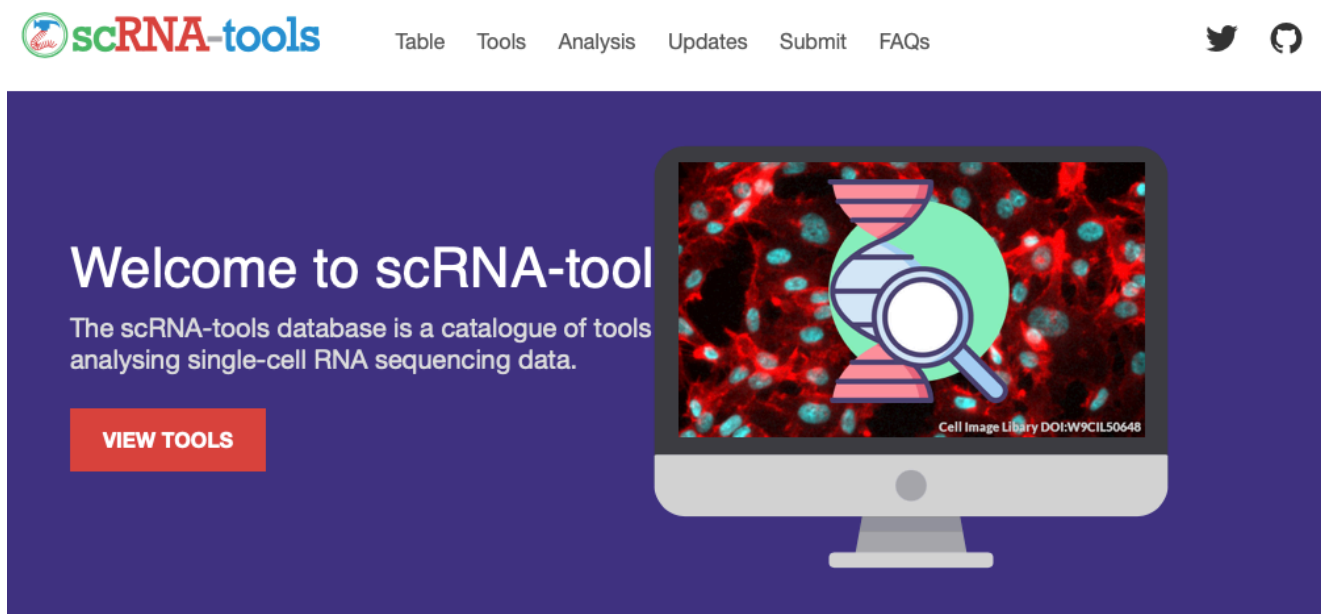
```
$ cd python_bioinfo_2023
```

```
$ jupyter notebook
```

Jupyter notebook で

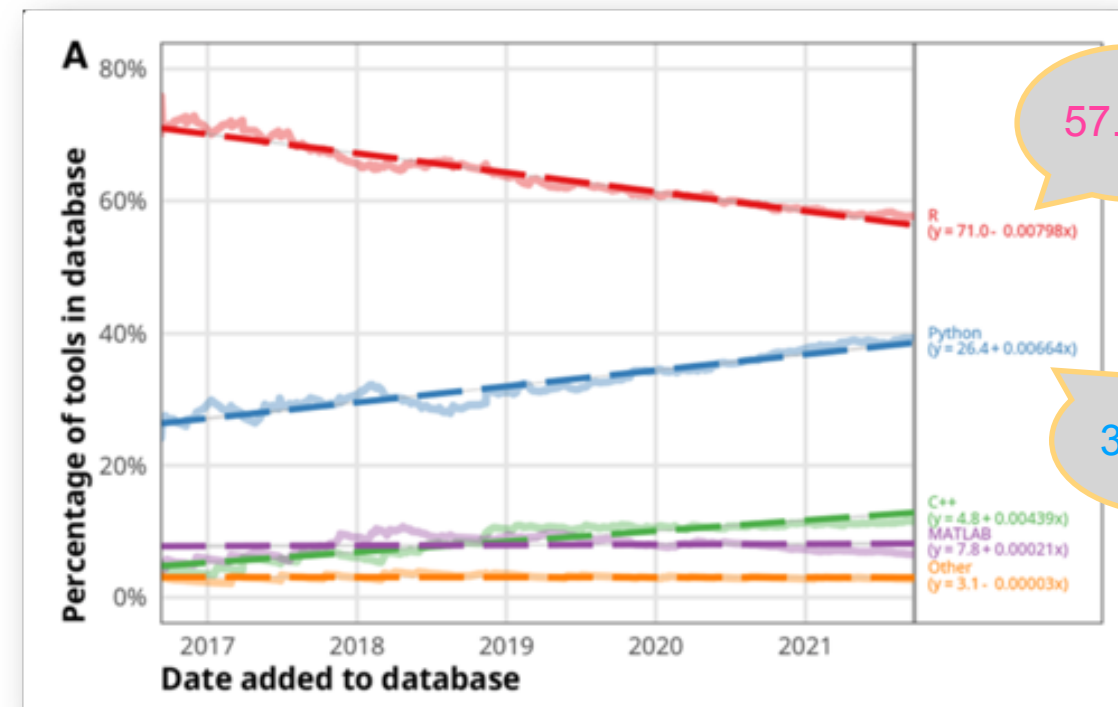
「2-3/Python_scRNAseq_1.ipynb」を起動してください。

PythonによるシングルセルRNA-seq解析



<https://www.scrna-tools.org/>

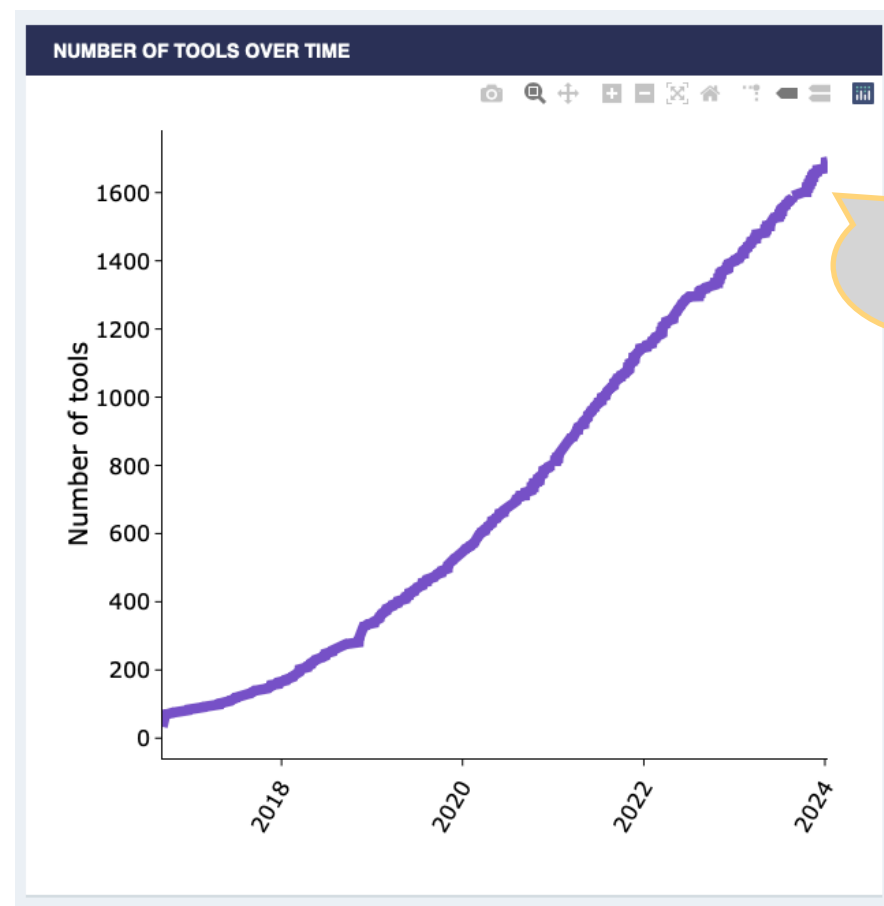
2024年1月16日現在



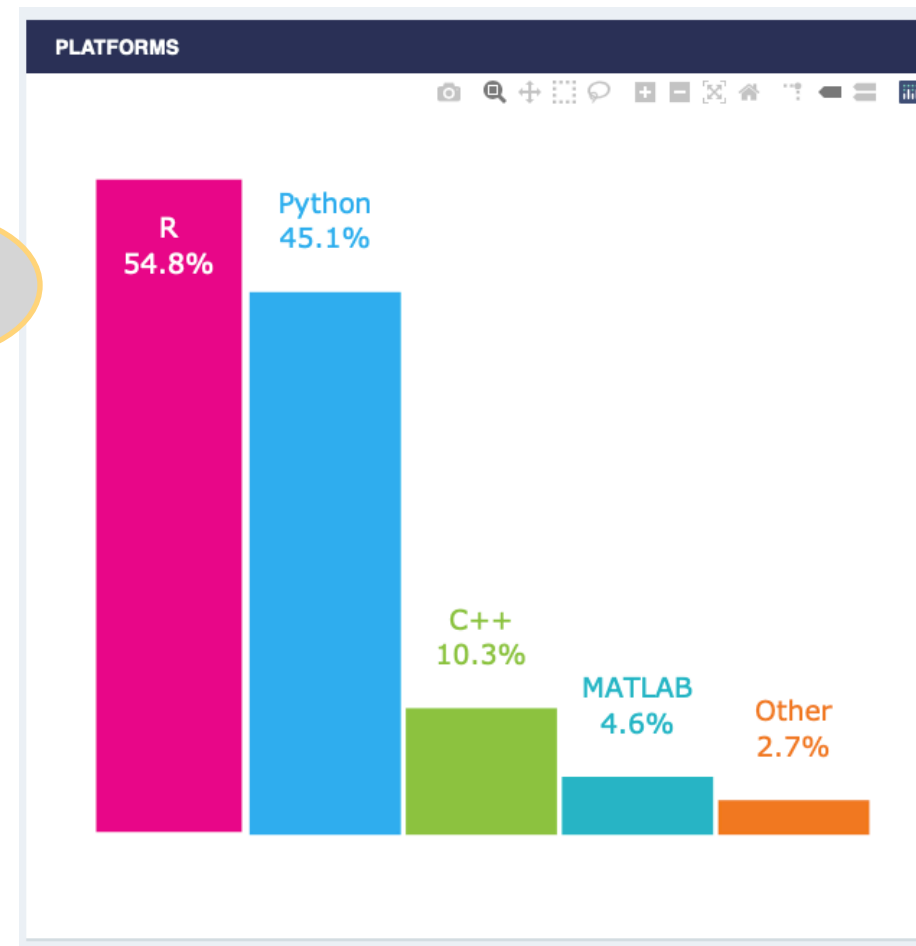
57.7 %

39.3 %

Zappia, L., Theis, F.J. *Genome Biol* **22**, 301 (2021).



1691 tools



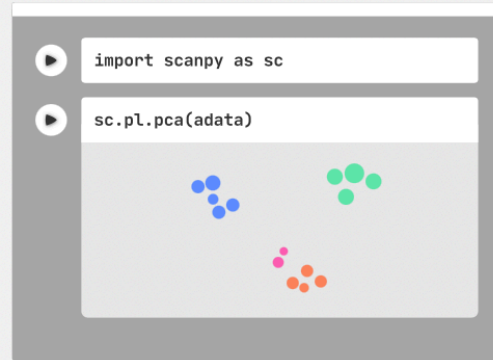
scverse 単一細胞オミクス解析に関連するPythonツールの開発・維持のためのコンソーシアム

<https://scverse.org>

scverse

Foundational tools for single-cell omics data analysis

[GitHub](#) [Discourse](#) [Zulip](#) [Twitter](#) [YouTube](#)



Virshup I. et al. *Nature Biotechnology* **41**, 604 (2023)

CORE PACKAGES



anndata

Standard for annotated matrices



mudata

Multimodal data format



scanpy

Single-cell analysis framework



muon

Multi-omics analysis framework



scvi-tools

Single-cell machine learning framework



scirpy

Single-cell immune sequencing analysis framework



squidpy

Spatial single cell analysis

AnnDataとScanpyをコア技術とする。

マルチモーダルデータ（scRNA-seq + scATAC-seq）の解析に対する
拡張として MuData, Muon の開発、

空間トランスクリプトーム解析のための Squidpy の開発など。

それぞれの相互運用性の改善やファイルフォーマットの統一など、
一体として扱いやすいツール群の開発を目指していくコミュニティ。

本日の講習

Cell ranger にて遺伝子毎のリードカウントを計算する



前処理: 細胞、遺伝子のフィルタリング



正規化



特徴量選択 (発現量の変動が大きい遺伝子)



Scanpy

主成分分析 (PCA)



次元削減 (UMAP, tSNE)



Leiden クラスタリング

計算

効率のために PCA
を行ってから UMAP,
tSNE を行う。

scRNA-seq データの統合 (バッチ補正)

scVI



バッチ補正後の UMAP~ Leiden クラスタリング

Scanpy



Doublet 検出

Solo



DEG解析

scVI

Scanpy: Pythonでシングルセル解析をする際のコアパッケージ



The screenshot shows the Scanpy website. On the left is a dark sidebar with the Scanpy logo (an orange fish-like shape) and the word "scanpy" in white. Below the logo is a "stable" badge and a "Search docs" input field. The sidebar menu includes: Tutorials, Usage Principles, Installation, API, External API, Ecosystem, Release notes, Community, and News. The main content area has a header "Scanpy – Single-Cell Analysis in Python" with an "Edit on GitHub" link. Below the header is a row of badges: Stars (1.7k), pypi (v1.9.6), downloads (2M), downloads (113k), docs (passing), Azure Pipelines (succeeded), discourse (3.7k posts), zulip, and join chat. The main heading is "Scanpy – Single-Cell Analysis in Python". The text below states: "Scanpy is a scalable toolkit for analyzing single-cell gene expression data built jointly with [anndata](#). It includes preprocessing, visualization, clustering, trajectory inference and differential expression testing. The Python-based implementation efficiently deals with datasets of more than one million cells." A list of links follows: Discuss usage on [Discourse](#) and development on [GitHub](#); Get started by browsing [tutorials](#), [usage principles](#) or the main [API](#); Follow changes in the [release notes](#); Find tools that harmonize well with anndata & Scanpy via the [external API](#) and the [ecosystem page](#); Check out our [contributing guide](#) for development practices; Consider citing [Genome Biology \(2018\)](#) along with original [references](#). On the right, a "Key Contributors" section lists: [anndata graph](#) | [scanpy graph](#) | ✨ = maintainer; [Isaac Virshup](#): lead developer since 2019 ✨; [Gökçen Eraslan](#): developer, diverse contributions ✨; [Sergei Rybakov](#): developer, diverse contributions ✨; [Fidel Ramirez](#): developer, plotting ✨.

10x のデータの場合、リード処理と定量化を Cell Ranger で行い、その後の処理を Scanpy で行う。

データの前処理や、近傍グラフ構築、UMAP, t-SNEなど、標準的な解析を実行できる。

基本的に、

AnnDataオブジェクトを入力して関数を実行すると、
結果が同じAnnDataオブジェクトに追加されていく。

新しいAnnDataを返すのではなく、
inplaceで（＝破壊的に）AnnDataが変換されていくのが特徴。

一見どこにどんな変化が生じたのかわかりにくい。

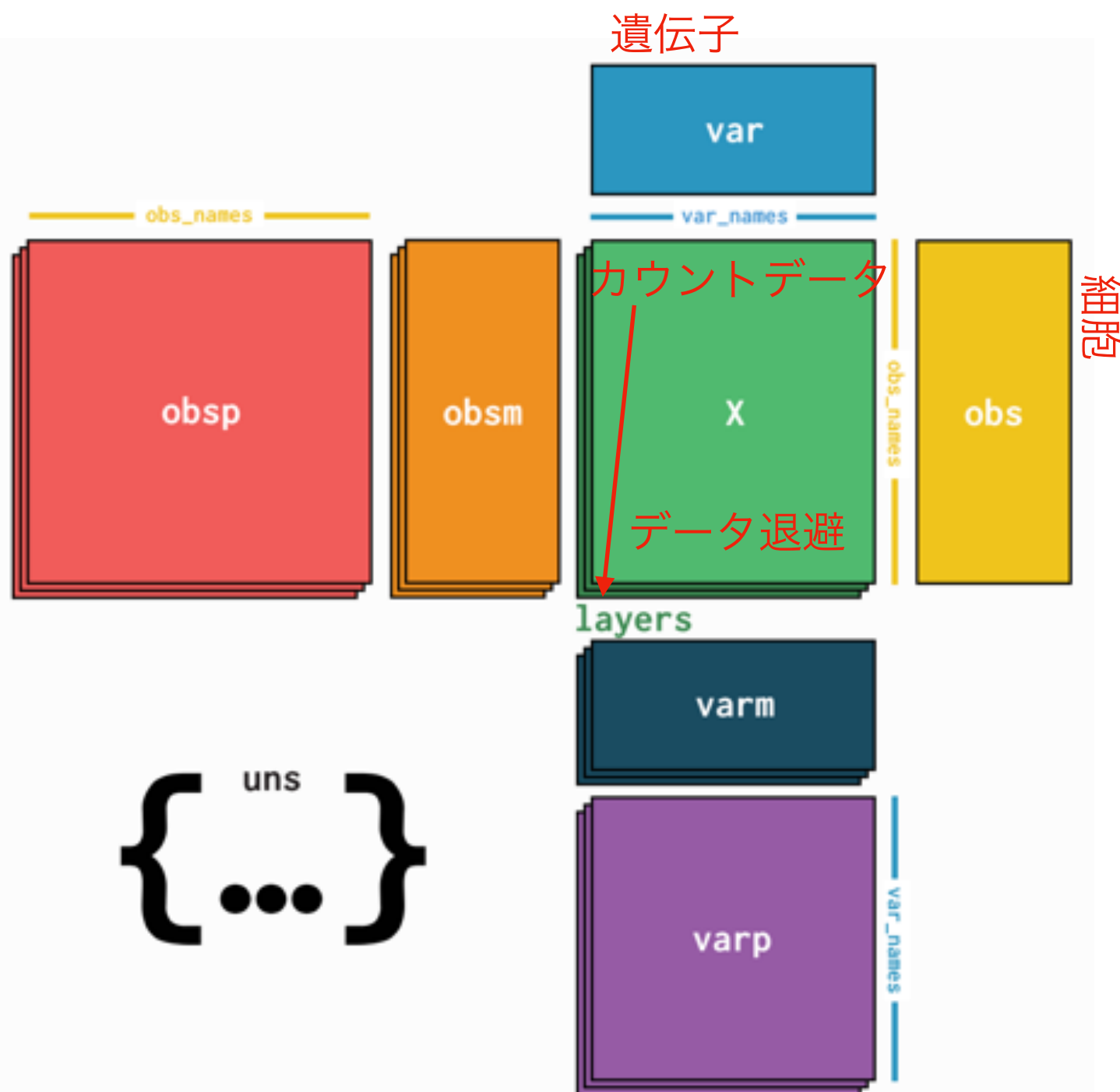
観測値や変数のデータフレームにいつのまにか勝手にカラムが追加されていることがある。

(“Annotated Data” の略)

シングルセル解析のためのPythonパッケージの多くが、このオブジェクトを使用

PandasのDataFrameを拡張したデータ構造

すべての観測と計算結果をひとつのオブジェクトに詰め込んで管理しやすくしたのが、AnnData というオブジェクトの特徴。



ひとつのオブジェクトに遺伝子発現量のデータ、サンプルや細胞のアノテーション、遺伝子の情報などをまとめて格納できる。

anndataを使うことで、実験の情報が詰まったひとつのオブジェクトに対して処理を次々に実行し、さらに処理結果をそこに蓄積していくことができる。

•.X

$n_obs \times n_vars$ の数値テーブル。numpy.ndarrayやscipyのスパースマトリックス。scRNA-seqのカウントマトリックスなど、実験の根幹となるデータ。

layers に、同じshapeの複数のマトリックスを保持しておける。たとえば全体をノーマライズしたけど元々のカウントデータも残しておきたいときは別のレイヤーに入れておく。スライスの影響はすべてのlayerに作用する。

•.obs

observationsの略。観測値に関するメタデータ。PandasのDataFrameなのでPandasの操作は全部実行できる。長さは必ず n_obs

•.var

variablesの略。変数（遺伝子など）に関するメタデータ。PandasのDataFrame。長さは必ず n_vars

•.obsm

multi-dimensional annotations for obs. 複数の数値のまとまりでそれぞれの観測値を表現したいときに使う。各観測値の低次元空間座標など。PCA で次元削減したデータもここに入る。次元サイズは任意。 $n_obs \times$ 次元サイズの numpy.ndarray.

•.varm

multi-dimensional annotations for var. $n_var \times$ 次元サイズのnumpy.ndarray

•.obsp

Pairwise annotation of obs. 観測値のペアに関する情報。距離行列など。 $n_obs \times n_obs$ のnumpy.ndarray

•.varp

Pairwise annotation of var. 変数のペアに関する情報。距離行列など。

$n_var \times n_var$ のnumpy.ndarray

•.uns

それ以外のデータ。とくに構造の制限はない。その他の関連データをひとまとめにしておきたいときに辞書型で放り込んでおく。クラスタの色指定とか。

Scanpyの関数

•scanpy.pp.XXX

前処理（preprocessing）に関連する関数がある。

細胞や遺伝子のフィルタリング、対数変換や、近傍グラフの構築など

•scanpy.tl.XXX

さまざまなツール（tools）のセット。

PCA, t-SNE, UMAP などの次元削減や、Louvain / Leiden クラスタリングなど。

•scanpy.pl.XXX

プロット（plotting）用の関数。

PCA用のプロット、UMAP用のプロットなど、それぞれの可視化に適した関数が用意されている。

複雑な処理を書かなくても、`anndata`に含まれるメタデータから自動的に、

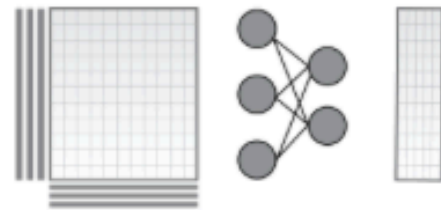
遺伝子発現量による色のグラデーションや、クラスタごとの色分けなどをしてくれる。

注：scanpyはたいてい“sc”の短縮名で呼び出すことが多いので、以上の関数は、`sc.pp.XXX`, `sc.tl.XXX`などと呼び出す

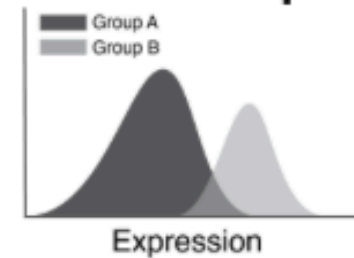
scvi-tools: 深層生成モデルを利用したシングルセルデータの確率的解析

複雑な確率モデルを
ニューラルネット
ワークを使って表現

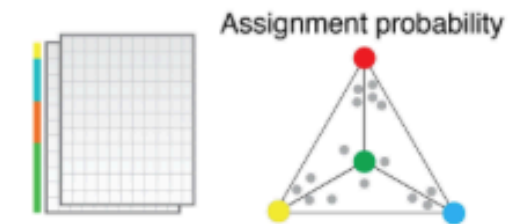
Dimensionality reduction



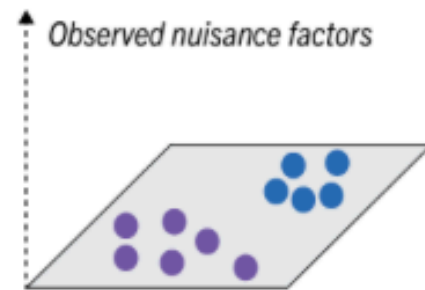
Differential comparison



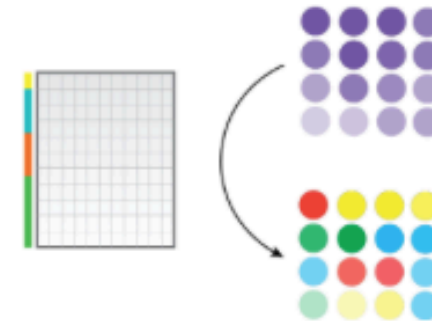
Automated annotation



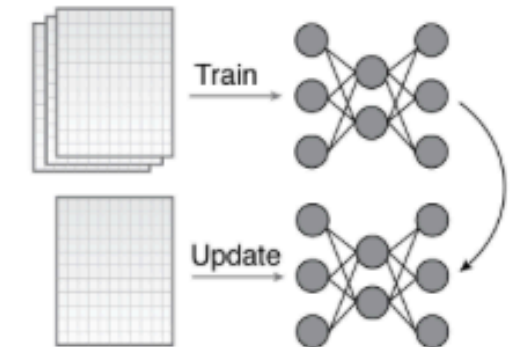
Removal of unwanted variation



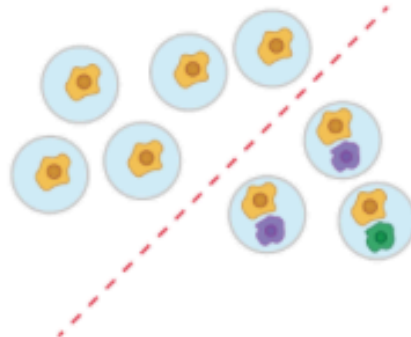
Deconvolution



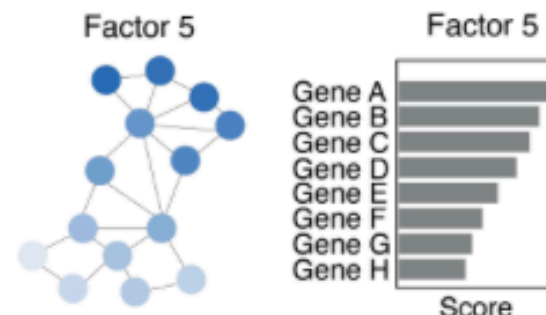
Transfer learning



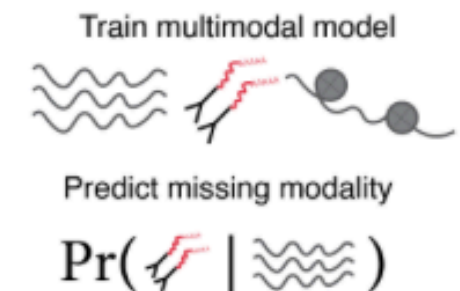
Doublet detection



Factor analysis



Modality imputation



次元削減

正規化

Replicate のデータ統合

インプテーション

などの機能が実装されている。

Gayoso, Adam, et al. "A Python library for probabilistic analysis of single-cell omics data."
Nature Biotechnology 40.2 (2022): 163-166.

Cell Systems

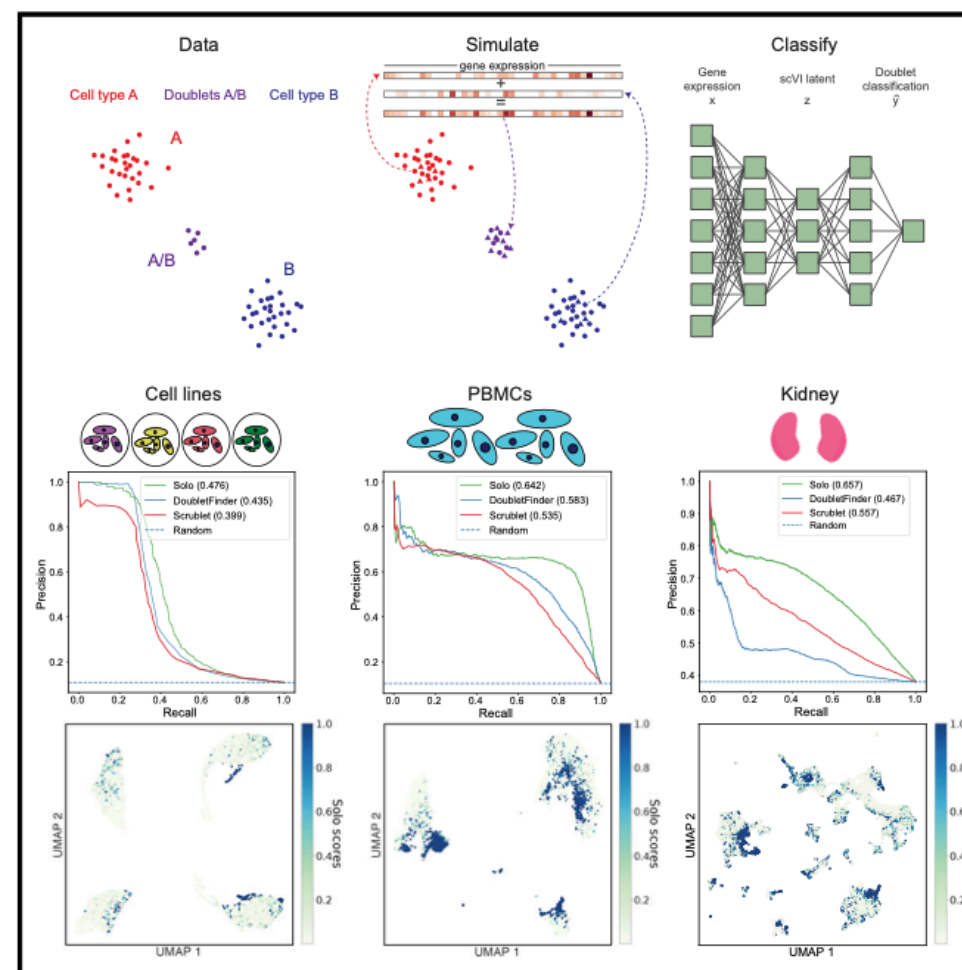
Solo: Doublet Identification in Single-Cell RNA-Seq via Semi-Supervised Deep Learning

scVIでモデリングした変分オートエンコーダの構造を流用。

エンコーダ（カウントデータから潜在表現への変換）の出力部分に、single/doubletの二分類を予測するニューラルネットワークを接続。

シミュレーションデータ（適当なふたつの細胞の平均発現パターン）でニューラルネットワークを学習してから、実際のデータのダブルットを予測する。

Graphical Abstract



Authors

Nicholas J. Bernstein, Nicole L. Fong, Irene Lam, Margaret A. Roy, David G. Hendrickson, David R. Kelley

Correspondence

dgh@calicolabs.com (D.G.H.), drk@calicolabs.com (D.R.K.)

In Brief

Current single-cell RNA sequencing technologies occasionally allow multiple cells to be combined into a single profile, which challenges downstream analyses. Bernstein et al. introduce a semi-supervised deep learning method called Solo that identifies these “doublet” cells with greater accuracy than existing methods.

Bernstein, Nicholas J., et al. *Cell systems* 11.1 (2020): 95-101.