Seurat, Signacを使ったシングルセル解析

Seurat, Signacはそれぞれ、scRNA-seq, scATAC-seq結果からシングルセル解析を行うツールです。いずれもRのパッケージとして提供されています。今回は10x genomics社のChromiumプラットフォームで取得したサンプルについてCellranger、Cellranger-ATACで処理した結果を用いて、Seurat、Signacを使った統計解析を行います。

解析に使用するデータについて

- 今回は、マウスの肺細胞について10x GenomicsのChromiumプラットフォームを用いて得られたscRNA-seq及びscATAC-seq配列について、10x GenomicsのCell Ranger及びCell Ranger-ATACというツールを用いて処理した結果を用いて解析を行います。
- Chromium結果の配列ファイルはDBKEROの
  - https://kero.hgc.jp/cgi-bin/download/tutorials/learning/data/10X_RNAv3_Lung.tar
  - https://kero.hgc.jp/cgi-bin/download/tutorials/learning/data/10X_ATAC_Lung.tar

  からダウンロードできます。

※ Cell Rangerの実行は時間が掛かりますので今回は割愛します。

※ Cell Rangerによる解析方法については、
https://kero.hgc.jp/open_tutorials/learning/contents/single1_article.htmlにまとめられています。

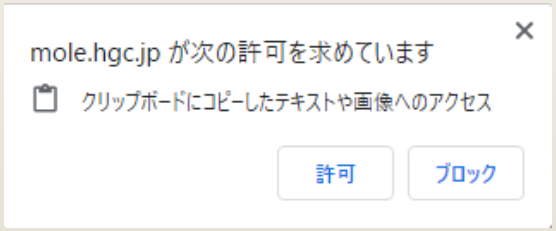
- Cell Rangerの結果についてはDBKEROの
  - https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scRNA (scRNA-seq)
  - https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC (scATAC-seq)

  以下に保存されています。

  今回はCell Rangerの結果を用いて解析の演習を行います。

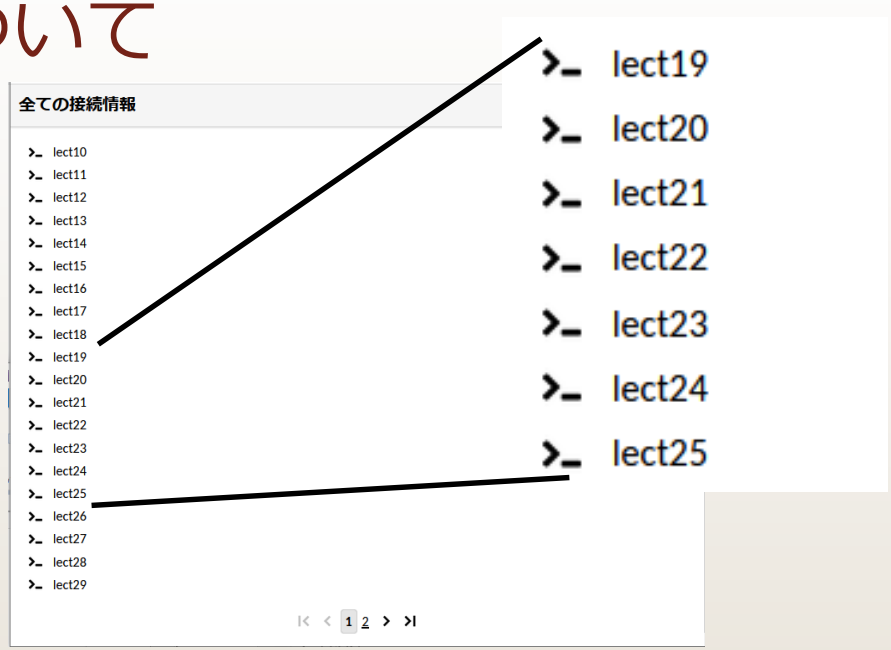
Guacamoleシステムを使ったSHIROKANEサーバへのSSHログインについて

- Guacamole (ワカモレ)は Web ブラウザから利用できるSSHクライアントです。
  Guacamoleシステムの詳細は https://supcom.hgc.jp/internal/mediawiki/id/1355 にあります。
  (※ 閲覧にはSHIROKANEユーザアカウントとパスワードが必要です)
- 受講者の方々には、今回こちらのシステムを利用して操作していただきます。（他のsshクライアントを用いたログインでも同様に操作できますが、今回の講習ではGuacamoleをご利用ください)
- Firefoxではマウスでコピペができませんでした。ChromeやEdgeブラウザのご使用をお願いします。
- GuacamoleシステムのURLは https://mole.hgc.jp/guacamole/#/ です。
- ユーザ名・パスワードは、お知らせするものをご利用ください。



- 上記の画面が出たら**[許可]**を選択して、javascriptのクリップボードの使用を許可してください！

Guacamoleシステムを使ったSHIROKANEサーバへのSSHログインについて



- 上記のような画面が現れたら、ご自身のユーザ名(lectXX)をクリックしてください。
- 他のユーザをクリックした場合もログインできてしまいますので、**必ずご自身のユーザでログイン**するようにしてください。他のユーザでログインしてしまった場合はexit後、ホームに戻って操作をやり直してください。

Guacamoleシステムを使ったSSHの利用

- 以下のようなコマンドライン画面が現れ、コマンドが実行できるようになります。

```
Last login: Fri Dec  4 16:23:50 2020 from 202.175.157.98

[Begin first by QLOGIN!]
Welcome to SHIROKANE supercomputer of Human Genome Center, IMS, UT.
This is a login node as the gateway with very little software.
You can type  qlogin  COMMAND to use all the features of SHIROKANE.
    |         o                ,,
,--.| ,--.,--.. ,--. |  --, / +-   -+-    .
|  || |  ||  || |  | |   / /  |-- | ` |  |
`--|`-`--'`--|`-`  ` `-- `-`- '`- `-  `-
    `      `--' https://supcom.hgc.jp/internal/mediawiki/id/357
[A login node of SHIROKANE lect10@slogin2:~]$ qlogin
要求ハードリソース
  memory (s_vmem): 4.5G = ジョブは 1 スロットあたり 4.5G バイトのメモリを要求します
  slots (def_slot): 1 = ジョブは 100% の CPU を要求します
  total memory: 4.5G = ジョブは 4.5G バイトのメモリを要求します
通常の qlogin を実行します。
Your job 28326615 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 28326615 has been successfully scheduled.
Establishing /home/geadmin/N1GE/utilbin/lx-amd64/qlogin_wrapper session to host gc016i ...
Warning: Permanently added '[gc016i]:35735,[172.28.1.24]:35735' (ECDSA) to the list of known hosts.
Last login: Fri Dec  4 15:00:53 2020 from yc001i
[OS 7] You are now on OS 7 compute node.
==== あなたのグループ lect のリソース利用状況 ====  hauq command version 1.12
* Home Disk use> 2 TB / 96 TB (2.1 %)   642 kfiles / 96000 kfiles (0.7 %)
* Arch Disk use> 0.0 TB / 0 TB (- %)   [ 0.0 TB(cache) + 0.0 TB(tape) ]
* UGE queue use> mjobs.q: 0/4096 (0 %)  ljobs.q: 0/768 (0 %)  lmem.q: 0/128 (0 %)  intr.q: 1/96 (1 %)
[lect10@gc016 ~]$
```

解析データの準備

- Seuratの解析にはCell Rangerの3つの結果ファイル、Signacの解析にはCell Ranger-ATACの4つの結果ファイルが必要です。
- お手元のPCのコマンドシェルからスパコンSHIROKANEにログイン・qloginし、以下のようにディレクトリを構成しデータをダウンロードしてください。

```
$ qlogin -l s_vmem=14G,mem_req=14G ⏎
…
$ #データDL用ディレクトリの作成とSeuratに使うデータのダウンロード ⏎
$ wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/1_cp.sh ⏎
$ sh 1_cp.sh ⏎
…
$
$ cat 1_cp.sh ⏎
#!/usr/bin/env sh

#Seuratに使うデータのDL
mkdir -p ~/public_html/analysis_scRNA/emm_rnaseq/outs/filtered_feature_bc_matrix/
cd ~/public_html/analysis_scRNA/emm_rnaseq/outs/filtered_feature_bc_matrix/
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scRNA/emm_rnaseq/outs//filtered_feature_bc_matrix/matrix.mtx.gz
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scRNA/emm_rnaseq/outs/filtered_feature_bc_matrix/barcodes.tsv.gz
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scRNA/emm_rnaseq/outs/filtered_feature_bc_matrix/features.tsv.gz

#Signacに使うデータのDL
mkdir -p ~/public_html/analysis_scATAC/emm_atac/outs/
cd ~/public_html/analysis_scATAC/emm_atac/outs/
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC/emm_atac/outs/filtered_peak_bc_matrix.h5
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC/emm_atac/outs/singlecell.csv
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC/emm_atac/outs/fragments.tsv.gz
wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC/emm_atac/outs/fragments.tsv.gz.tbi

$
```

解析データの準備

- データのダウンロード後、以下のように正しくファイルがダウンロードされているか確認してください。

```
$ #ホームディレクトリに戻る ⏎
$ cd ⏎
$ ls -l ~/public_html/analysis_scRNA/emm_rnaseq/outs/filtered_feature_bc_matrix/ ⏎
合計 134516
-rw-r----- 1 lectXX lect  19360229 11月 13 20:00 barcodes.tsv.gz
-rw-r----- 1 lectXX lect    260010 11月 13 20:01 features.tsv.gz
-rw-r----- 1 lectXX lect 118114730 11月 13 20:00 matrix.mtx.gz
$ ls -l ~/public_html/analysis_scATAC/emm_atac/outs/ ⏎
合計 1292832
-rw-r----- 1 lectXX lect   50333876 11月 15 10:38 filtered_peak_bc_matrix.h5
-rw-r----- 1 lectXX lect 1240316124 11月 15 10:39 fragments.tsv.gz
-rw-r----- 1 lectXX lect     765761 11月 15 10:39 fragments.tsv.gz.tbi
-rw-r----- 1 lectXX lect   32435230 11月 15 10:38 singlecell.csv
$
```

解析データの準備

- Cell Ranger(上)/Cell Ranger ATAC(下)の結果出力ファイルの内容は以下の通りです。

File Name	Description
web_summary.html	HTML形式のサマリー。
metrics_summary.csv	CSV形式のサマリー。
possorted_genome_bam.bam	BAMファイル、IGVで閲覧可能。
possorted_genome_bam.bam.bai	BAIファイル。
filtered_feature_bc_matrix	細胞ごとの各遺伝子のUMI数のデータ。filter（UMI数の閾値）をパスした細胞のデータのみ。
filtered_feature_bc_matrix.h5	HDF5形式。細胞ごとの各遺伝子のUMI数のデータ。filterをパスした細胞のみ。
raw_feature_bc_matrix	細胞ごとの各遺伝子のUMI数のデータ。Filterではじかれたデータも含む。
raw_feature_bc_matrix.h5	HDF5形式。細胞ごとの各遺伝子のUMI数のデータ。Filterではじかれたデータも含む。
analysis	DEGや、クラスタリング結果の情報を含むディレクトリ。
molecule_info.h5	Cell Rangerの再解析に使用する。
cloupe.cloupe	Loupe Cell Browser用のファイル。

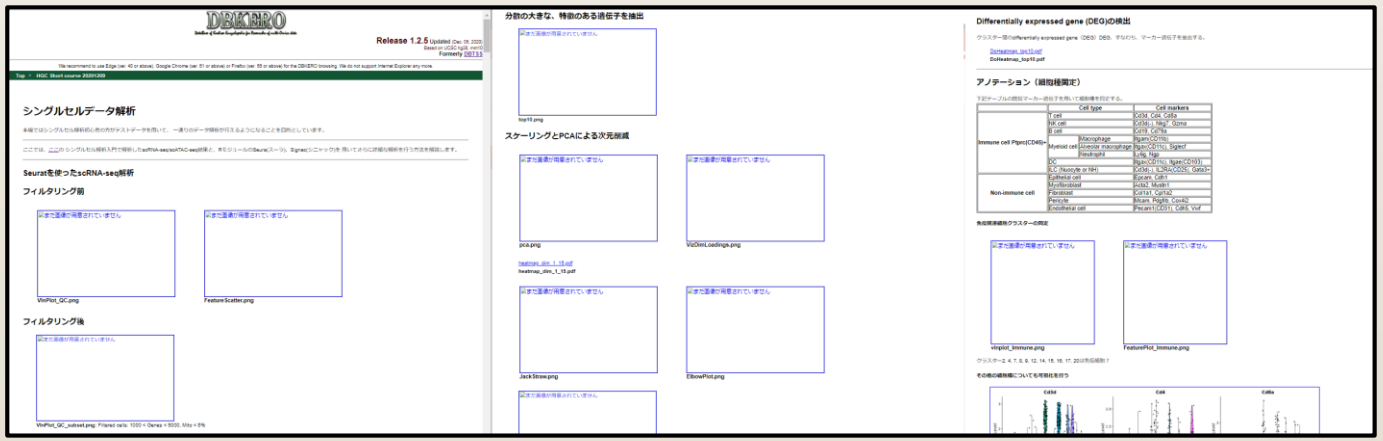
File Name	Description
singlecell.csv	バーコードフラグメントごとのカウントマトリクスデータ
possorted_bam.bam	ソート済みbamファイル
possorted_bam.bam.bai	上記のindexファイル
summary.json	JSON形式のサマリー
web_summary.html	HTML形式のサマリー
peaks.bed	ピーク領域データ
raw_peak_bc_matrix.h5	hdf5形式のピークデータ
raw_peak_bc_matrix	ピークデータ
analysis	解析結果を含むディレクトリ
filtered_peak_bc_matrix.h5	hdf5形式のフィルタ後のピークマトリックス
filtered_peak_bc_matrix	フィルタ後のピークマトリックスデータ
fragments.tsv.gz	バーコードフラグメント位置情報のタブ区切りファイル
fragments.tsv.gz.tbi	上記のtabixファイル
filtered_tf_bc_matrix.h5	hdf5形式の細胞ごとの転写因子のデータ
filtered_tf_bc_matrix	細胞ごとの転写因子のデータ
cloupe.cloupe	Loupe Cell Browser用のファイル
summary.csv	CSV形式のサマリー。

結果表示用HTMLの準備

- 解析で得られた画像ファイルを閲覧したりダウンロードするために、以下の要領でHTMLファイルを用意し、ホームディレクトリにpublic_htmlディレクトリを作成し、その中に配置します。
- public_html/ 以下は https://www.hgc.jp/~lect**XX**/ 以下からブラウザで閲覧できます(lectXX はご自身のユーザ名に置き換えてください)。

```
$ #Web公開ディレクトリの権限変更 ⏎
$ chmod 755 ~/public_html/ ⏎
$ chmod 755 ~/public_html/* ⏎
$ cd ~/public_html/ ⏎
$
$ #結果閲覧用HTMLのダウンロード
$ wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/hgc_short_course_20201209/index.html ⏎
$ #ファイル権限の変更
$ chmod 644 ~/public_html/index.html ⏎
```

- index.htmlをダウンロードしたらWebブラウザからページを確認できることを確認してください。



※ 「まだ画像が用意されていません」と表示されますが、解析が進むと画像が生成されていきます
※ 講習で作成したHTMLは**講習後は閲覧できなくなります**ので予めご了承ください

解析ツールの準備

- Seurat、Signacの解析方法については、https://kero.hgc.jp/open_tutorials/learning/contents/single2_article.html にまとめています。
- Seurat、Signacを利用するためには多くのCモジュールや、Rモジュールのインストールが必要です。
- スムーズに進んでもこれらのインストールには何時間か掛かりますので、今回は既にSeurat、Signacをインストールした状態のSingularity コンテナイメージを予め用意しておきました。今回はこのイメージファイルをダウンロードし、利用することにします。

```
$ #SHIROKANE上で適切なバージョンのSingularityを利用できる状態にし、Singularityコンテナを起動する ⏎
$ cd ⏎
$ wget https://kero.hgc.jp/cgi-bin/download/tutorials/learning/3_execute_singularity.sh ⏎
$ sh 3_execute_singularity.sh ⏎
…
Singularity centos8_20201110.sif:~>
```

- 上記のように「Singularity centos8_20201110.sif:~>」と表示されれば成功です
- シェルスクリプトの内容は以下のようになっています。

```
$ cat 3_execute_singularity.sh ⏎
#!/usr/bin/env sh

#SHIROKANE上で適切なバージョンのSingularityを利用できる状態にする
module load /usr/local/package/modulefiles/singularity/3.2.1

#Singularityのバージョンが 3.2.1-1になっていることを確認します。
singularity ―version

#Singularityコンテナイメージをダウンロードします。
wget ―no-clobber https://kero.hgc.jp/cgi-bin/download/tutorials/learning/centos8_20201110.sif

#コンテナを起動します。
singularity shell centos8_20201110.sif
```

解析ツールの準備

- 続いてRを起動し、Seuratと、Signacのバージョンが正しく表示されることを確認してください

```
Singularity centos8_20201110.sif:~> #Rの実行とSeurat、Signacのバージョン確認 ⏎
Singularity centos8_20201110.sif:~> R ⏎

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
…
…
Type 'q()' to quit R.

> packageVersion("Seurat") ⏎
[1] '3.2.2'
> packageVersion("Signac") ⏎
[1] '1.1.0'
> q() ⏎
Save workspace image? [y/n/c]: n ⏎
Singularity centos8_20201110.sif:~>
```

※ Rはq()を入力すると終了することができます。imageの保存は"n"で問題ありません。yにすると、保存に時間が掛かるので注意！

Seuratを使ったscRNA-seq解析

このセクションの資料は以下のページを参考に作成しています。

https://satijalab.org/seurat/v3.2/pbmc3k_tutorial.html (2020.10.02版)

処理手順は、データのフィルタリング(精度の良い細胞データの抽出) → 解析に使用する遺伝子の選抜→ PCAによるデータの次元圧縮→クラスタリングとUMAPによる可視化→ DEG検出および、各クラスターのマーカー遺伝子の発現情報から、細胞種とクラスターの対応付けという順番で解析を進めます。

データの確認と解析準備

- SeuratはscRNA-seq解析によく利用されている解析ツールです。Rのモジュールとして提供されています。ホームページが https://satijalab.org/seurat/ にありますので、詳細はここをご確認ください。
- 最初に作業ディレクトリを作成しRを起動します。

```
$ mkdir -p ~/public_html/analysis_scRNA/Seurat
$ cd ~/public_html/analysis_scRNA/Seurat #解析ディレクトリを作成し移動
$ #chmod 755 ~/public_html/analysis_scRNA/Seurat #公開のためパーミッションを変更
$
$ R #Rを起動します
```

- 今回は既にCell Rangerで処理済みの結果を使って処理を行います。
- 以下のように lung.dataにCell Rangerの結果(細胞ごとの各遺伝子のUMI数のデータ)を読み込み、Seuratオブジェクトlungを生成します。

```
> #解析スタート
> library(Seurat)
> library(dplyr)
> set.seed(1234)
>
> lung.data <- Read10X(data.dir =
            "~/public_html/analysis_scRNA/emm_rnaseq/outs/filtered_feature_bc_matrix")
> lung <- CreateSeuratObject(counts = lung.data, project = "lung", min.cells = 3,
            min.features = 200)
```

※ lung.dataに読み込んだデータファイル(filtered_feature_bc_matrix)は、各行が遺伝子、各列が細胞のUMI値(=分子数)のカウントマトリクスとなっています
※ Seuratオブジェクトの詳細は https://github.com/satijalab/seurat/wiki をご参照ください

解析に使用する細胞のフィルタリング

- 精度関係なくすべての細胞を確認するとトータル8,214細胞が検出されました。ここから精度のよくない細胞を取り除くフィルタリング処理を行っていきます。

```
> #マトリクスデータに含まれる細胞数の確認
> length(lung@meta.data$nFeature_RNA)
[1] 8214
```

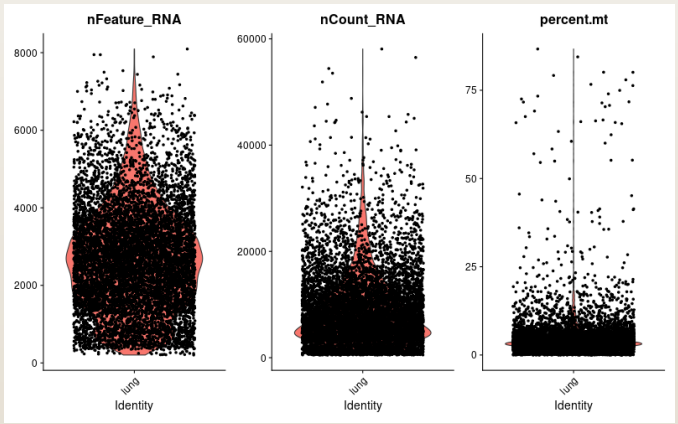
- 一般的にscRNA-seqのQCによく使われる指標としては以下の3つがあります。
  - 1. 各セルで検出される遺伝子の種類の数
    - 低品質の細胞や空の液滴には、遺伝子がほとんどないことがよくあります
    - 細胞ダブレットまたはマルチプレットは、逆に異常に高い遺伝子数を示す可能性があります
  - 2. 各細胞内で検出されたUMI数（1.と強く相関します）
  - 3. ミトコンドリアゲノムにマップされるリード数の割合
    - 低品質/死にかけている細胞は、しばしば広範なミトコンドリア汚染が見られます。PercentageFeatureSetという関数を用いることで、mt-で始まるすべての遺伝子セットを抽出することで、ミトコンドリア遺伝子由来リードの割合を取得することができます。
- 以下のようにPercentageFeatureSet を用いるとmeta.dataにミトコンドリアリードのデータが追記されます。

```
> lung[["percent.mt"]] <- PercentageFeatureSet(lung, pattern = "^mt-")
> #metadataの確認
> head(lung@meta.data)
                  orig.ident nCount_RNA nFeature_RNA percent.mt
AAACCCAAGCCTAGGA-1       lung      11245         3313   3.112494
AAACCCACAAGGATGC-1       lung      14923         4079   2.579910
AAACCCACAATTCGTG-1       lung      11329         3554   4.060376
AAACCCACACGAGGTA-1       lung       3779         1591   5.054247
AAACCCACAGAAGCTG-1       lung      11357         3619   2.694374
AAACCCACATGACAAA-1       lung       3686         1530   3.635377
```

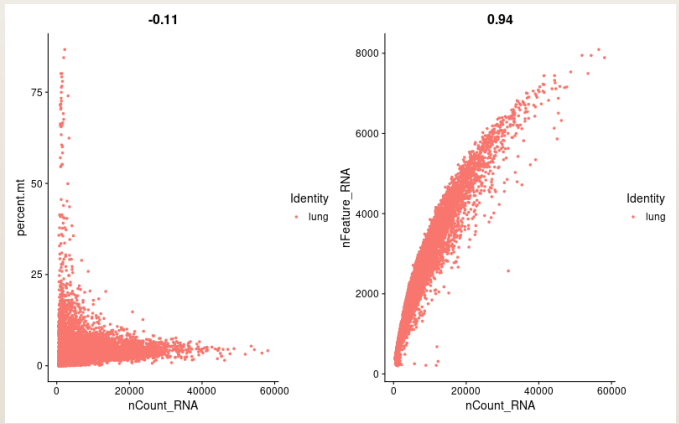
解析に使用する細胞のフィルタリング

- 遺伝子数、UMI数。ミトコンドリアリードの割合の分布をグラフにします

```
> png("VlnPlot_QC.png", width = 800, height = 500)
> VlnPlot(lung, features=c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
> dev.off()
> #nFeature_RNA: 検出遺伝子の種類の数, nCount_RNA: UMI数, percent.mt: ミトコンドリア割合
> png("FeatureScatter.png", width = 800, height = 500)
> plot1 <- FeatureScatter(lung, feature1 = "nCount_RNA", feature2 = "percent.mt")
> plot2 <- FeatureScatter(lung, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
> plot1 + plot2
> dev.off()
```



VlnPlot_QC.png

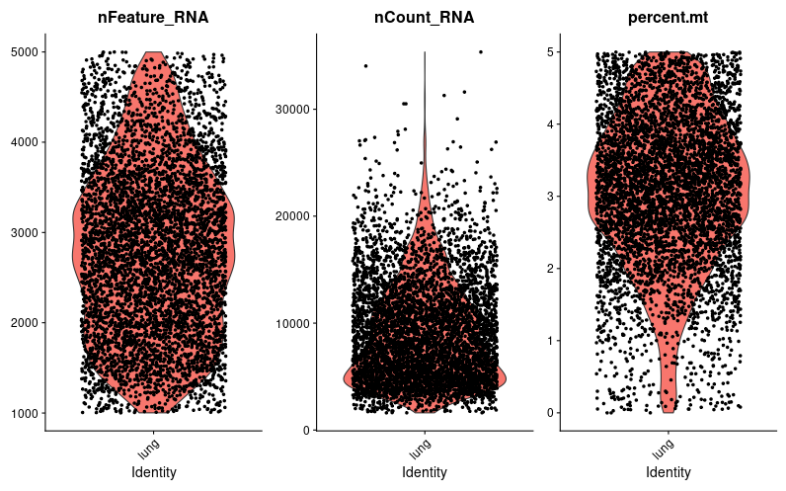


FeatureScatter.png

解析に使用する細胞のフィルタリング

- データ精度の良くない細胞を取り除きます

```
> #検出遺伝子数1000以上、5000以下、ミトコンドリア5%以下でフィルタリング（ここの値をどうするかが重要！）
> lung <- subset(lung , subset = nFeature_RNA > 1000 & nFeature_RNA < 5000 & percent.mt < 5)
> #検出遺伝子数およびUMI数が多い細胞はマルチプレットの可能性が高く除外する
> #ミトコンドリアは5-10%以下とすることが多い。細胞種によって異なるので、分布をきちんと確認し適切な閾値を設定する
> #細胞数の確認
> length(lung@meta.data$nFeature_RNA)
[1] 5136
> #細胞フィルタリング後のQCプロットを出力
> png("VlnPlot_QC_subset.png", width = 800, height = 500)
> #nFeature_RNA: 特徴量数(検出遺伝子数), nCount_RNA: UMIの数, percent.mt: ﾐﾄｺﾝﾄﾞﾘｱの割合
> VlnPlot(lung, features=c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
> dev.off()
```



VlnPlot_QC_subset.png:
Filtered cells: 1000 < Genes < 5000, Mito < 5%

解析に使用する細胞のフィルタリング

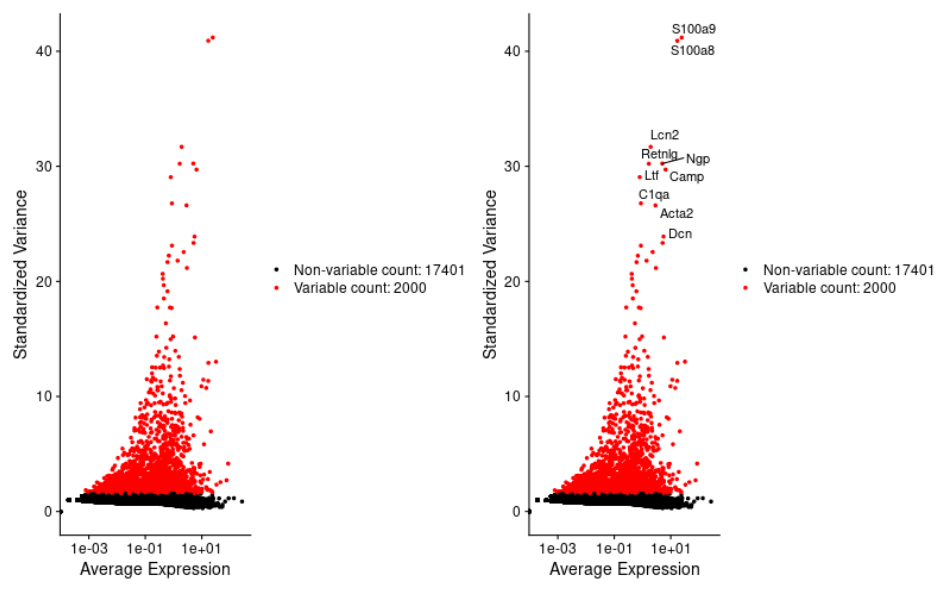
- 次にコンピュータが扱いやすい数値にするためデータの正規化を行います。今回は、各セルの遺伝子発現量をトータルの発現量で正規化し、これに係数（10,000）を掛けて、結果を対数変換します。正規化された値は lung[["RNA"]]@data に格納されます。

```
> #logでノーマライズする（scale.factor=10000 → 1万リードあたりのtag数に正規化(default)）
> lung <- NormalizeData(lung, normalization.method = "LogNormalize", scale.factor = 10000)
>
> #中身を確認
> head(lung[["RNA"]]@data, 1)
6 x 5136 sparse Matrix of class "dgCMatrix"
   [[ suppressing 5136 column names 'AAACCCAAGCCTAGGA-1', 'AAACCCACAAGGATGC-1',
'AAACCCACAATTCGTG-1' ... ]]

Xkr4    . . .         .         . . . . . .     . .         . .
Gm19938 . . .         .         . . . . . .     . .         . .
Rp1     . . .         .         . . . . . .     . .         . .
Sox17   . . .         .         . . . . . .     . 1.9810015 . .
Gm37587 . . .         .         . . . . . .     . .         . .
Mrpl15  . . 0.6327018 0.6315453 . . . . 0.984 . 0.8109302 . 0.4681139

...
>
```

解析に使用する遺伝子データの抽出

- 分散の大きな、発現に特徴のある遺伝子を抽出します (上位2,000個)。PCAの次元圧縮で使用されます。

```
> #分散の大きい(発現変動の大きい)2000遺伝子のみを抽出し、解析に使用
> lung <- FindVariableFeatures(lung, selection.method = "vst", nfeatures = 2000)
>
> #遺伝子の分散をグラフ上に可視化。top10についてラベルを付加
> top10 <- head(VariableFeatures(lung), 10)
> png("top10.png", width = 800, height = 500)
> plot1 <- VariableFeaturePlot(lung)
> plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
> plot1 + plot2
> dev.off()
```



Top10.png

※ 遺伝子抽出の手法については、https://www.biorxiv.org/content/biorxiv/early/2018/11/02/460147.full.pdf に説明がされているようです。

遺伝子発現データのスケーリングとPCAによる次元圧縮

- PCA計算のためのスケーリング (正規化)を行います。全細胞の平均発現量が0、細胞間の発現量の分散が1になるように各遺伝子の発現量をスケーリングします。

```
> all.genes <- rownames(lung)
> lung <- ScaleData(lung, features = all.genes)
> # lung <- ScaleData(lung)とすると、2000遺伝子のみをスケーリングするため短時間で済むが、
ヒートマップを描く際にスケーリングしていない遺伝子を入れる場合は再スケーリングする必要が出て
くる
```

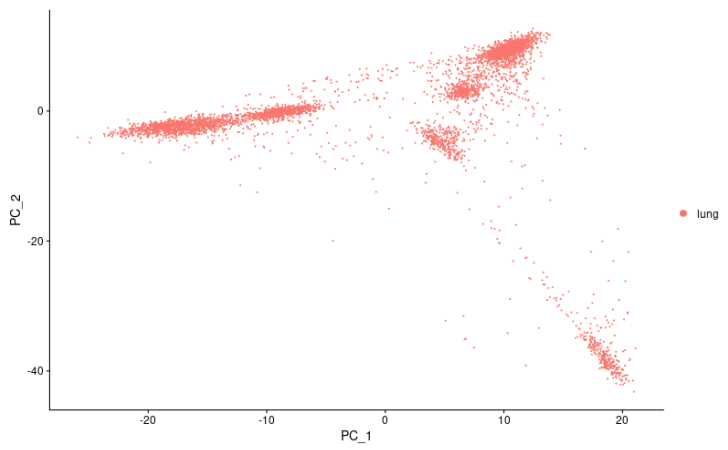
- PCAによる次元圧縮。今回は抽出した2000遺伝子を元に次元圧縮を行います。

```
> lung <- RunPCA(lung, features = VariableFeatures(object = lung))
> print(lung[["pca"]], dims = 1:5, nfeatures = 5)
PC_ 1
Positive:  Sparc, Igfbp7, Pmp22, Timp3, Ccn1
Negative:  Laptm5, Coro1a, Ptprc, Lcp1, Arhgdib
PC_ 2
Positive:  Serping1, Bgn, Col1a2, Sparcl1, Plpp3
Negative:  Aqp5, Rtkn2, Cyp2b10, Gprc5a, Clic3
PC_ 3
Positive:  Anxa1, Prdx6, S100a6, Gsn, Mettl7a1
Negative:  Cdh5, Cldn5, Ramp2, Egfl7, Tspan7
PC_ 4
Positive:  Ptprcap, Sept1, Ms4a4b, Trbc2, Lck
Negative:  Psap, Sirpa, Ccl6, Ear2, Lyz2
PC_ 5
Positive:  Slc43a3, Cfh, Gpc3, Fmo2, Scn7a
Negative:  Postn, Cox4i2, Higd1b, Ndufa4l2, Pdgfrb
>
```

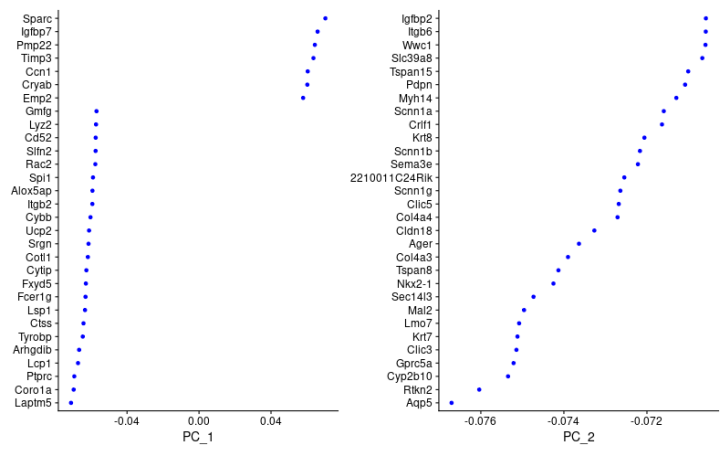
遺伝子発現データのスケーリングとPCAによる次元圧縮

- PCA結果のグラフ化 (DimPlot及びVizDimLoadings)

```
> png("pca.png", width = 800, height = 500)
> DimPlot(lung, reduction = "pca")
> dev.off()
>
> png("VizDimLoadings.png", width = 800, height = 500)
> VizDimLoadings(lung, dims = 1:2, reduction = "pca")
> dev.off()
```



PCA.png

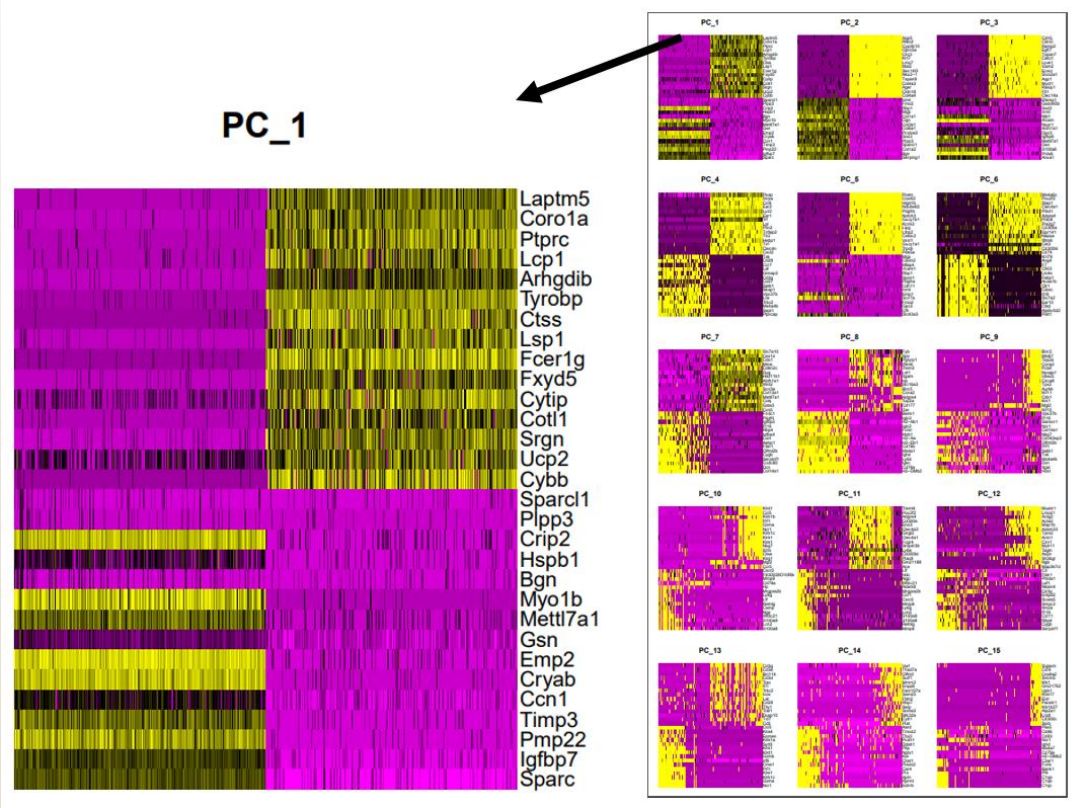


VizDimLoadings.png

遺伝子発現データのスケーリングとPCAによる次元圧縮

- 各細胞の遺伝子発現のヒートマップを生成 (デフォルトでは上位下位の15遺伝子が表示されるようです。横軸にはcells=500の場合上位250、下位250の細胞が選択されています)

```
> pdf("heatmap_dim_1_15.pdf", width = 8, height = 15)
> DimHeatmap(lung, dims = 1:15, cells = 500, balanced = TRUE)
> dev.off()
```

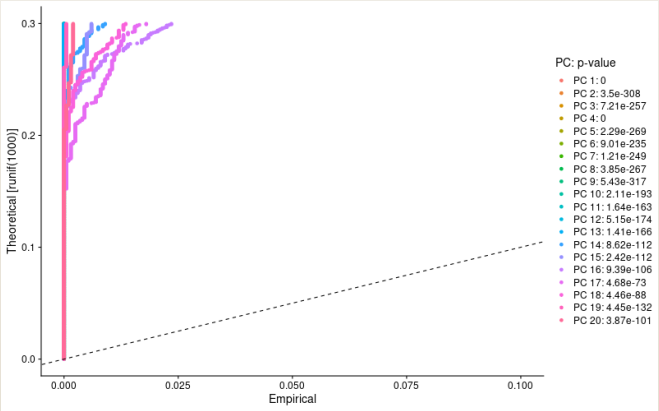


heatmap_dim_1_15.pdf

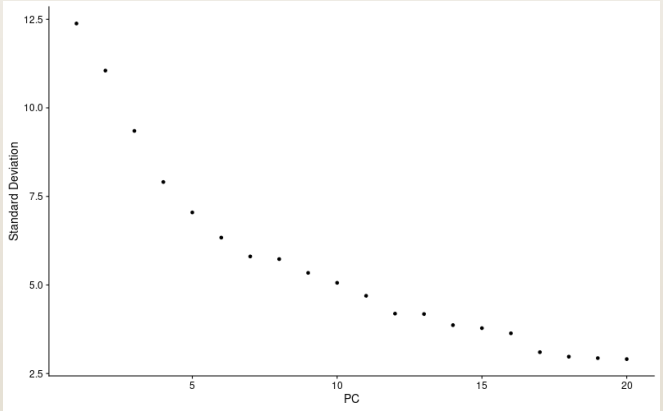
遺伝子発現データのスケーリングとPCAによる次元圧縮

- 以後の解析に用いる次元数を決めるため指標となるグラフを作成

```
> #少し時間がかかる。時間があればもっと次元数を増やしたほうが良い
> lung <- JackStraw(lung, num.replicate = 100, dims = 20)
> lung <- ScoreJackStraw(lung, dims = 1:20)
>
> png("JackStraw.png", width = 800, height = 500)
> JackStrawPlot(lung, dims = 1:20)
> dev.off()
>
> png("ElbowPlot.png", width = 800, height = 500)
> ElbowPlot(lung)
> dev.off()
```



JackStraw.png



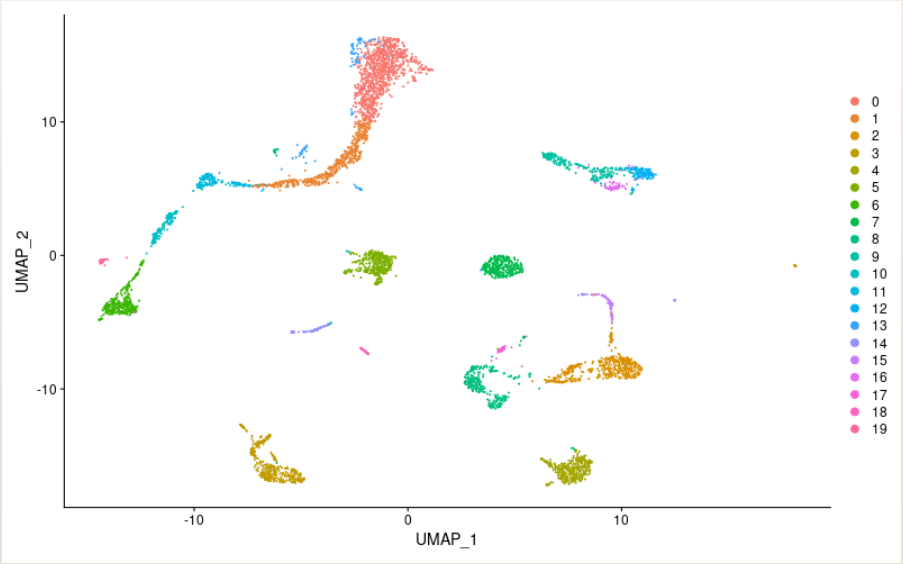
ElbowPlot.png

※ 理想的なグラフは https://satijalab.org/seurat/v3.2/pbmc3k_tutorial.html#determine-the-dimensionality-of-the-dataset を参考になさってください。

クラスタリングとUMAPによるプロット

- PC1-PC10の10個の主成分を用いてクラスタリングとUMAPでの可視化を行います。

```
> # クラスタリング
> lung <- FindNeighbors(lung, dims = 1:10)        #仮にPC1-10で解析を進めています
> lung <- FindClusters(lung, resolution = 0.5)
>
> # UMAPで二次元プロット
> lung <- RunUMAP(lung, dims = 1:10)              #tSNEが良い場合はRunTSNE関数を使います
>
> png("umap.png", width = 800, height = 500)
> DimPlot(lung, reduction = "umap")
> dev.off()
```



umap.png

※ いくつの次元までのPCを用いるか、決定するのが難しい場合も多いと思います。なるべく多くの次元を用いて処理し、より少ない次元でも結果があまり変わらない場合は下げていくという方法を取るとよいと思われます。

Differentially expressed gene (DEG)の検出

- クラスター間のdifferentially expressed gene（DEG）、すなわち、マーカー遺伝子を抽出します。
- SeuratオブジェクトのFindAllMarkersがこの一連の処理を行ってくれます。検出手法として以下のMethodが用意されています。

- **wilcox: Wilcoxon Rank Sum test (default)**
- bimod: Likelihood-ratio test
- roc: Standard AUC classifier
- t: Student's t-test
- poisson: Use only for UMI-based datasets
- LR: Uses a logistic regression framework
- negbinom: Use only for UMI-based datasets
- MAST: GLM-framework
- DESeq2

```
> #全クラスター
> lung.markers <- FindAllMarkers(lung, only.pos = TRUE, min.pct = 0.25, logfc.threshold =
0.25) #かなり時間がかかる。
> lung.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_logFC)
Registered S3 method overwritten by 'cli':
  method     from
  print.boxx spatstat
# A tibble: 40 x 7
# Groups:   cluster [20]
       p_val avg_logFC pct.1 pct.2 p_val_adj cluster gene
       <dbl>     <dbl> <dbl> <dbl>     <dbl> <fct>   <chr>
 1 0.             2.03 0.984 0.295 0.        0       Ces1d
 2 0.             1.97 0.998 0.536 0.        0       Mfap4
 3 1.61e-238      1.77 1     0.846 3.13e-234 1       Mgp
 4 5.80e-234      1.62 0.98  0.412 1.12e-229 1       Cxcl14
 5 0.             2.32 0.971 0.26  0.        2       Plac8
 6 0.             2.21 0.978 0.157 0.        2       Lst1
 7 0.             3.19 0.972 0.086 0.        3       Cldn5
 8 0.             3.16 0.992 0.171 0.        3       Ramp2
 9 0.             3.27 1     0.154 0.        4       Chil3
10 0.             2.78 0.954 0.17  0.        4       Tnf
# ... with 30 more rows
>
```

FindAllMarkersの引数について：

- only.pos = TRUE: positive マーカーのみを抽出する
- min.pct = 0.25: 2つの細胞グループのいずれかでpct(クラスタ内変動割合)が0.25以上として検出された遺伝子のみを処理する。pctの低い遺伝子を処理しないことで、処理をスピードアップする。
- Logfc.threshold = 0.25: 2つの細胞グループ間での発現比が平均で、ログスケールで少なくとも0.25倍の差がある遺伝子に絞って処理を行う。値を増やすと関数は高速になるが、低いシグナルを見逃す可能性がある。

Differentially expressed gene (DEG)の検出

- 個別に遺伝子セットを抽出したい場合の例

```
> #各クラスターの上位10マーカーを抽出
> top10 <- lung.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_logFC)
> #テキストに出力(top10のみ)
> write.table(top10, "top10.csv",append = FALSE, quote = TRUE, sep = "," ,row.names = TRUE, col.names = TRUE)
```

	A	B	C	D	E	F	G	H
1		p_val	avg_logFC	pct.1	pct.2	p_val_adj	cluster	gene
2	1	0	2.034322433	0.984	0.295	0	0	Ces1d
3	2	0	1.973771726	0.998	0.536	0	0	Mfap4
4	3	0	1.847289669	0.996	0.292	0	0	Cdo1
5	4	0	1.758687288	1	0.474	0	0	Gpx3
6	5	0	1.720580269	0.991	0.33	0	0	Tcf21
7	6	0	1.701885546	1	0.494	0	0	Hsd11b1
8	7	0	1.669858812	1	0.622	0	0	Inmt
9	8	0	1.656718575	0.999	0.474	0	0	Fmo2
10	9	0	1.649774431	0.991	0.307	0	0	Gsta3
11	10	0	1.649621281	0.999	0.38	0	0	Aldh1a1
12	11	0	1.445156475	0.982	0.261	0	1	Mfap5
13	12	0	1.421949631	0.853	0.108	0	1	C4b
14	13	1.61E-238	1.768413879	1	0.846	3.13E-234	1	Mgp
15	14	5.80E-234	1.623346668	0.98	0.412	1.12E-229	1	Cxcl14
16	15	6.30E-231	1.352499875	0.978	0.384	1.22E-226	1	Gas6
17	16	1.89E-224	1.274022455	0.932	0.331	3.67E-220	1	C3
18	17	1.20E-213	1.249526824	0.998	0.468	2.33E-209	1	Col3a1
19	18	1.60E-205	1.345842279	0.944	0.4	3.11E-201	1	Nbl1
20	19	8.62E-200	1.304739551	0.998	0.568	1.67E-195	1	Timp3
21	20	8.88E-154	1.281050872	0.845	0.335	1.72E-149	1	Maff
22	21	0	2.317290382	0.971	0.26	0	2	Plac8

top10

Top10.csv

Differentially expressed gene (DEG)の検出

- 各クラスターで個別にマーカー遺伝子を抽出する場合の例です

```
> # 各クラスターで個別に解析した場合：
> # クラスター1での発現変動遺伝子の同定
> cluster1.markers <- FindMarkers(lung, ident.1 = 1, min.pct = 0.25)
> #クラスター5とクラスター0,3間での発現変動遺伝子
> cluster5.markers <- FindMarkers(lung, ident.1 = 5, ident.2 = c(0, 3), min.pct = 0.25)
```

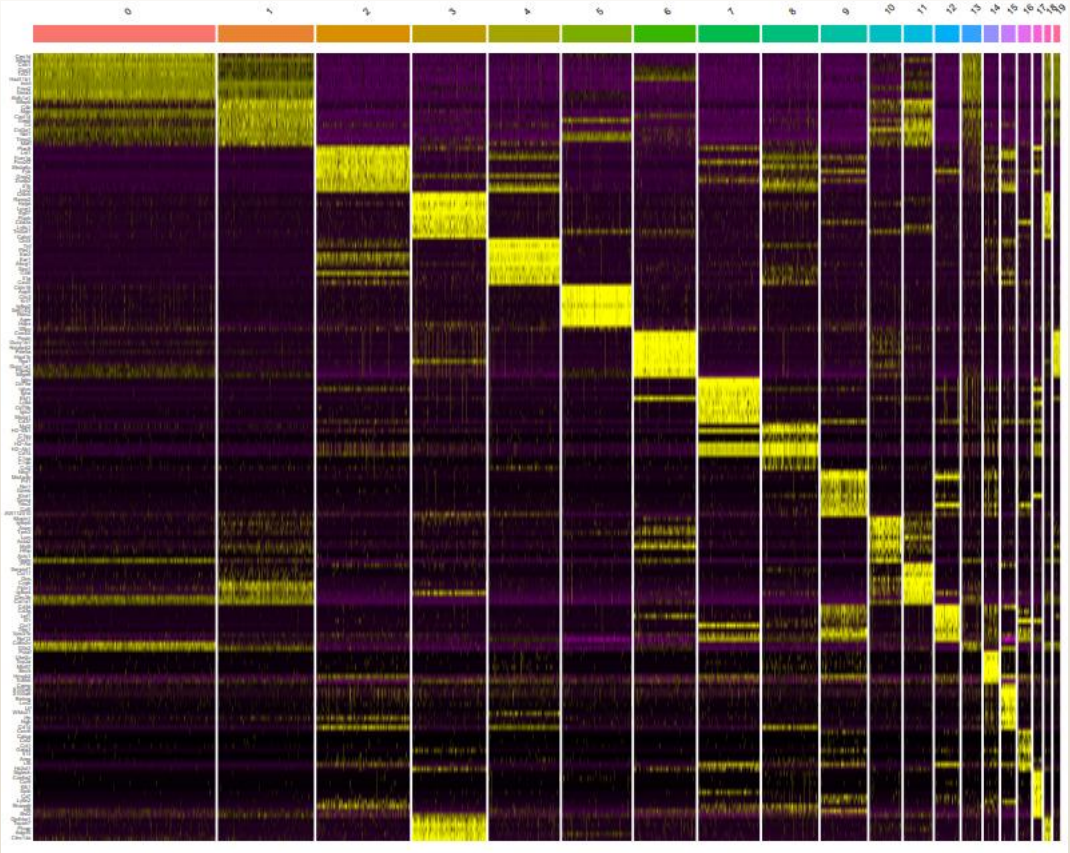
- 上位の行を表示

```
> head(cluster1.markers)
 p_val avg_logFC pct.1 pct.2 p_val_adj
Mfap5         0 1.4451565 0.982 0.261         0
C4b           0 1.4219496 0.853 0.108         0
Clec11a       0 1.0409024 0.722 0.092         0
Cygb          0 0.8913736 0.763 0.084         0
C2            0 0.8197373 0.755 0.101         0
Pdgfrl        0 0.7835708 0.811 0.115         0
>
> head(cluster5.markers)
 p_val avg_logFC pct.1 pct.2     p_val_adj
Nkx2-1    0.000000e+00  1.829041 0.948 0.026  0.000000e+00
Scnn1g   1.280513e-306  1.574700 0.904 0.016 2.484324e-302
Ptprf    3.660918e-304  1.621070 0.926 0.028 7.102546e-300
Cyp2b10  6.706375e-304  2.178700 0.931 0.033 1.301104e-299
Il18r1   1.213560e-303  1.564943 0.918 0.024 2.354428e-299
Pdpn     4.103163e-302  2.091630 0.953 0.046 7.960547e-298
```

Differentially expressed gene (DEG)の検出

- 各クラスターの上位10マーカーのヒートマップ図を出力します。

```
> pdf("DoHeatmap_top10.pdf", width = 25, height = 20)
> DoHeatmap(lung, features = top10$gene) + NoLegend()
> dev.off()
```



DoHeatmap_top10.pdf

アノテーション （細胞種同定）

- 下記テーブルの既知マーカー遺伝子を用いて細胞種を同定します。

	Cell type		Cell markers
Immune cell Ptprc(CD45)+	T cell (T細胞)		Cd3d, Cd4, Cd8a
	NK cell (NK細胞)		Cd3d(-), Nkg7, Gzma
	B cell (B細胞)		Cd19, Cd79a
	Myeloid cell (骨髄細胞)	Macrophage	Itgam(CD11b)
		Alveolar macrophage	Itgax(CD11c), Siglecf
		Neutrophil (好中球)	Ly6g, Ngp
	DC (樹状細胞)		Itgax(CD11c), Itgae(CD103)
	ILC (Nuocyte or NH) (自然リンパ球)		Cd3d(-), IL2RA(CD25), Gata3+
Non-immune cell	Epithelial cell (上皮細胞)		Epcam, Cdh1
	Myofibroblast (筋繊維芽細胞)		Acta2, Mustn1
	Fibroblast (繊維芽細胞)		Col1a1, Cpl1a2
	Pericyte (周皮細胞)		Mcam, Pdgfrb, Cox4i2
	Endothelial cell (内皮細胞)		Pecam1(CD31), Cdh5, Vwf

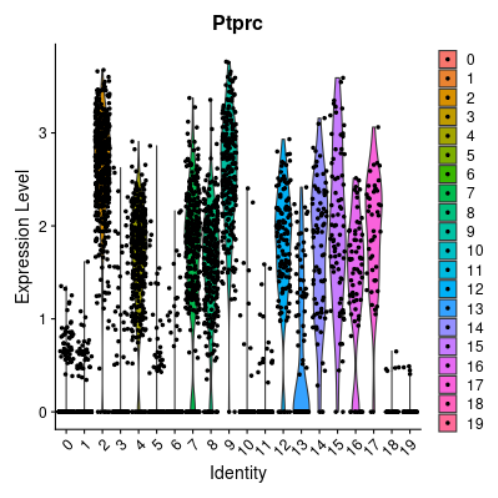
- 上表を元に、マーカーとなる遺伝子を変数に代入します。

```
> Immune <- c("Ptprc") #CD45
> T <- c("Cd3d", "Cd4", "Cd8a")
> NK <- c("Nkg7", "Gzma")
> B <- c("Cd19", "Cd79a")
> Myeloid <- c("Itgam", "Cd68")
> AlveolarMacrophage <- c("Itgax", "Siglecf", "Cd68")
> DC <- c("Itgax", "Itgae")
> Neutrophil <- c("Ly6g", "Ngp")
> Epithelial <- c("Epcam","Cdh1")
> Myofibroblast <- c("Mustn1","Acta2")
> Fibroblast <- c("Col1a1", "Col1a2")
> Endothelial <- c("Pecam1", "Cdh5", "Vwf")
> Pericyte <- c("Mcam", "Pdgfrb", "Cox4i2")
```

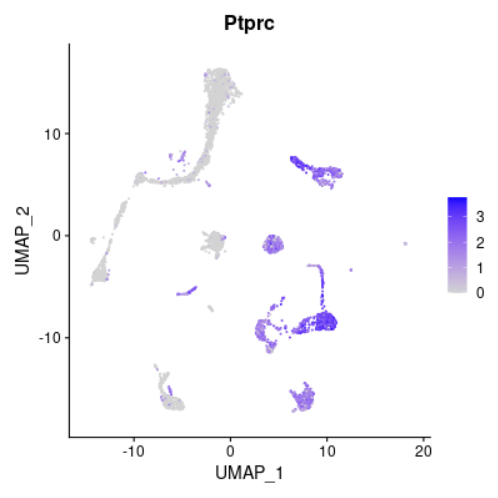
アノテーション （細胞種同定）

- 免疫関連細胞クラスター(Immune)の同定

```
> #バイオリンプロット(Immune cell)
> png("vlnplot_Immune.png", width=400, height=400)
> VlnPlot(lung, features = Immune)
> dev.off()
>
> #マーカー遺伝子の発現量でUMAPプロットを色塗り(Immune cell)
> png("FeaturePlot_Immune.png", width=400, height=400)
> FeaturePlot(lung, features = Immune)
> dev.off()
```



vlnplot_Immune.png



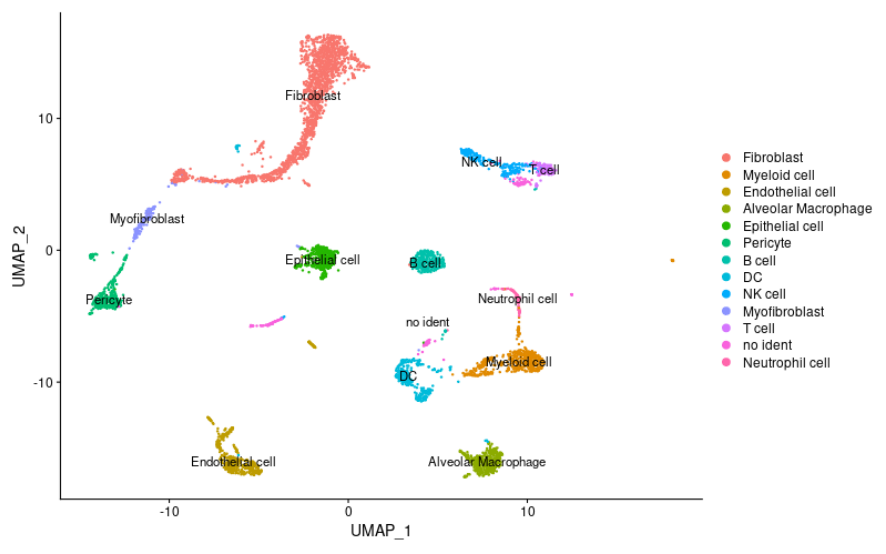
FeaturePlot_Immune.png

- 同様に features = Immune の部分を変えてグラフを作成します。作成した図は https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scRNA/Seurat/scRNA_annotation.zip からダウンロードできます。

アノテーション （細胞種同定）

- マニュアルで割り当てた細胞種をUMAP上に可視化します

```
> #クラスターへの細胞種の割り当て
> new.cluster.ids <- c("Fibroblast", "Fibroblast", "Myeloid cell", "Endothelial cell",
"Alveolar Macrophage", "Epithelial cell", "Pericyte", "B cell", "DC", "NK cell",
"Myofibroblast", "Fibroblast", "T cell", "Fibroblast", "no ident", "Neutrophil cell",
"no ident", "no ident", "Endothelial cell", "Pericyte", "no ident")
>
> names(new.cluster.ids) <- levels(lung)
> lung <- RenameIdents(lung, new.cluster.ids)
>
> #ラベル付きでUMAPプロットを出力
> png("umap_cls.png", width = 800, height = 500)
> DimPlot(lung, reduction = "umap", label = TRUE, pt.size = 0.5)
> dev.off()
```



- 以下のようにlungオブジェクトを必ず保存しておいてください。あとでscATAC-Seqとの統合解析に使用します

```
> #lungオブジェクトをファイルに保存します
>
> saveRDS(lung, file="lung_rna.rds")
>
> #Rを終了させます（保存はnにして下さい!）
> q()
```

FeaturePlot_Immune.png

Signacを使ったscATAC-seq解析

このセクションの資料は以下のページを参考に作成しています。

https://satijalab.org/signac/articles/pbmc_vignette.html (2020.11.06版)

処理手順は、データのフィルタリング(精度の良い細胞データの抽出) → 抽出データの次元圧縮 → クラスタリングとUMAPによる可視化となります。可視化後、scRNA-seqのクラスターに対してscATAC-seqで得られたクラスターを比較し、細胞種の対応付けを行います。

データの確認と解析準備

- SignacはscATAC-seq用の解析ツールです。Seuratと同じグループにより開発されているSeuratの拡張Rモジュールです。https://satijalab.org/signac/index.html に公開されています。
- 最初に作業ディレクトリを作成しRを起動します。

```
$ mkdir -p ~/public_html/analysis_scATAC/Signac #解析ディレクトリを作成
$ cd ~/public_html/analysis_scATAC/Signac
$ R
```

- まず以下のように、解析に必要なライブラリをロードします。

```
> #解析スタート
> #必要なライブラリを読み込む
> library(Signac)
> library(Seurat)
> library(GenomeInfoDb)
> library(EnsDb.Mmusculus.v79)
> #library(EnsDb.Hsapiens.v75) #humanの場合
> library(ggplot2)
> library(scales)
> library(patchwork)
> set.seed(1234)
```

データの確認と解析準備

- Cell Ranger ATACのアウトプットのうち下記4ファイルを使用します。
  - •filtered_peak_bc_matrix.h5
  - •singlecell.csv
  - •fragments.tsv.gz
  - •fragments.tsv.gz.tbi
- 以下のように、データを読み込み、SeuratObject (lung) を生成します。

```
> #入力ファイル読み込み
> counts <- Read10X_h5(filename =
"~/public_html/analysis_scATAC/emm_atac/outs/filtered_peak_bc_matrix.h5") #peakのファイル
>
> #データ読み込み
> metadata <- read.csv(file =
"~/public_html/analysis_scATAC/emm_atac/outs/singlecell.csv", header = TRUE, row.names =
1)
>
> #fragmentファイル読み込み
> chrom_assay <- CreateChromatinAssay(counts = counts, sep = c(":", "-"), genome = 'mm10',
  fragments = '~/public_html/analysis_scATAC/emm_atac/outs/fragments.tsv.gz', min.cells =
1, min.features = 200)
>
> #Seuratオブジェクトの生成
> lung <- CreateSeuratObject(counts = chrom_assay, assay = 'peaks', project = 'ATAC',
meta.data = metadata)
```

※4ファイルのうち、上の2つはピーク及び細胞のマトリクスファイルです。scRNA-seqで使用したカウントマトリクスに対応します。ただし遺伝子の代わりにゲノムのピーク領域となります。データの詳細は、https://support.10xgenomics.com/single-cell-atac/software/pipelines/latest/output/matrices で確認できます。下の2つはフラグメントファイルです。全シングルセルについての全リードのリストです。データの詳細は、https://support.10xgenomics.com/single-cell-atac/software/pipelines/latest/output/fragments で確認できます。

解析に使用する細胞のフィルタリング

- 以下のような指標に従って、解析に使用する細胞を選択していきます
  - ヌクレオソームバンドパターン： DNAフラグメントサイズのヒストグラム（ペアエンドシーケンシングリードから決定）は、シングルヌクレオソームに巻き付いたDNAの長さに強く依存したパターンを示します。シングルセルごとにこれを計算し、ヌクレオソームを含まないフラグメントに対するモノヌクレオソームのおおよその比率（nucleosome_signal）から外れるものを除外します。
  - ピーク中にアサインされるフラグメントの総数：細胞ごとのsequencing depth/complexity の値を利用する。 この値が小さい場合は、リード数が少ない細胞を示し、この値が非常に大きい細胞は、ダブレット等のアーティファクトを表す場合があります。
  - ピーク内にアサインされるフラグメントの割合：ATAC-seqピーク内にあるすべてのフラグメントの割合を表します。値が低い15～20%未満の細胞は、多くの場合、低品質の細胞またはアーティファクトを表し、フィルターする必要があります。この値は、使用されるピークのセットに影響される可能性があることに注意してください。
  - ブラックリスト領域にあるリードの比率: ENCODEプロジェクトでは、アーティファクトシグナルであることの多いリードを表すブラックリスト領域のリストを提供しています (https://github.com/Boyle-Lab/Blacklist)。これらの領域への読み取りマッピングの割合が高いセルは、多くの場合、アーティファクトを表しているため、フィルターする必要があります。

解析に使用する細胞のフィルタリング

- まず、フィルタリングや解析に用いる遺伝子アノテーションの情報をlungオブジェクトに追加します

```
> #lungに遺伝子アノテーション情報を追加(CoveragePlotに遺伝子アノテーションを表示するのに必要)
> annotations <- GetGRangesFromEnsDb(ensdb = EnsDb.Mmusculus.v79)
> #UCSC mm10にマップされた情報のため、Ensembl から UCSCスタイルのアノテーションに変更
> seqlevelsStyle(annotations) <- 'UCSC'
> genome(annotations) <- "mm10"
> #オブジェクトにこのannotation情報を追加
> Annotation(lung) <- annotations
```

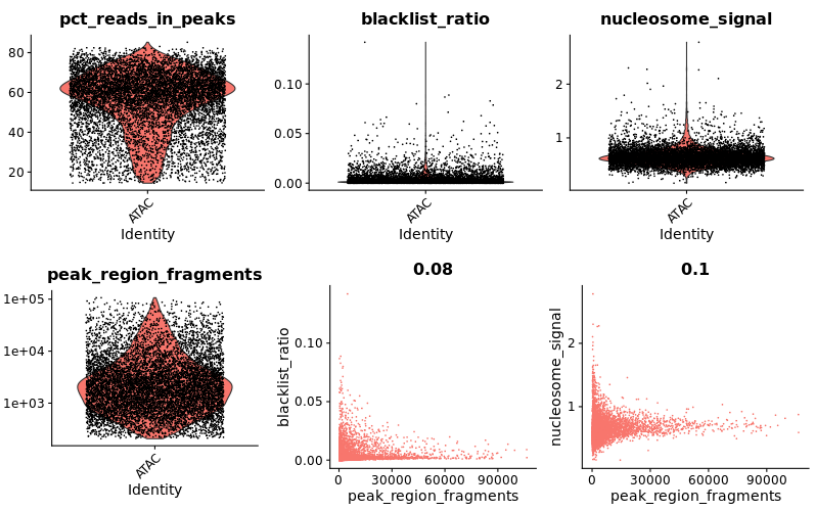
- 続いてフィルタリングを行います。・Nucleosome Signal(ヌクレオソームを1つ含むリードペアと、含まないリードペアの割合)と、・ピーク位置にアサインされるフラグメントの割合、・ブラックリスト領域にアサインされたリード割合、でフィルタリングを行います。

```
> # QC & フィルタリング
> #(mononucleosomal fragments)/(nucleosome-free fragments)
> lung <- NucleosomeSignal(object = lung, region = 'chr1-1-10000000')
>
> #ピークにアサインされたリードの割合 (%)
> lung$pct_reads_in_peaks <- lung$peak_region_fragments / lung$passed_filters * 100
>
> #blacklist (artificial signalが多い場所)にアサインされたリードの割合
> lung$blacklist_ratio <- lung$blacklist_region_fragments / lung$peak_region_fragments
```

解析に使用する細胞のフィルタリング

- ・ピーク位置にアサインされるフラグメント割合(pct_reads_in_peaks)、・ブラックリスト領域にアサインされたリード割合(blacklist_ratio)、 ・Nucleosome Signal、・ピーク中にあるフラグメントの総数(peak_region_fragments)をグラフにします。

```
> png("VlnPlot_FeatureScatter.png", width = 800, height = 500)
> plot1 <- VlnPlot(object = lung,features = c('pct_reads_in_peaks', 'blacklist_ratio',
'nucleosome_signal'),pt.size = 0.1) + NoLegend()
> plot2_a <- VlnPlot(object = lung,features = 'peak_region_fragments',pt.size = 0.1, log
= TRUE) + NoLegend()
> plot2_b <- FeatureScatter(lung,"peak_region_fragments",'blacklist_ratio', pt.size =
0.1) + NoLegend()
> plot2_c <- FeatureScatter(lung,"peak_region_fragments",'nucleosome_signal', pt.size =
0.1) + NoLegend()
> plot2 <- plot2_a | plot2_b | plot2_c
> plot1 / plot2
> dev.off()
```

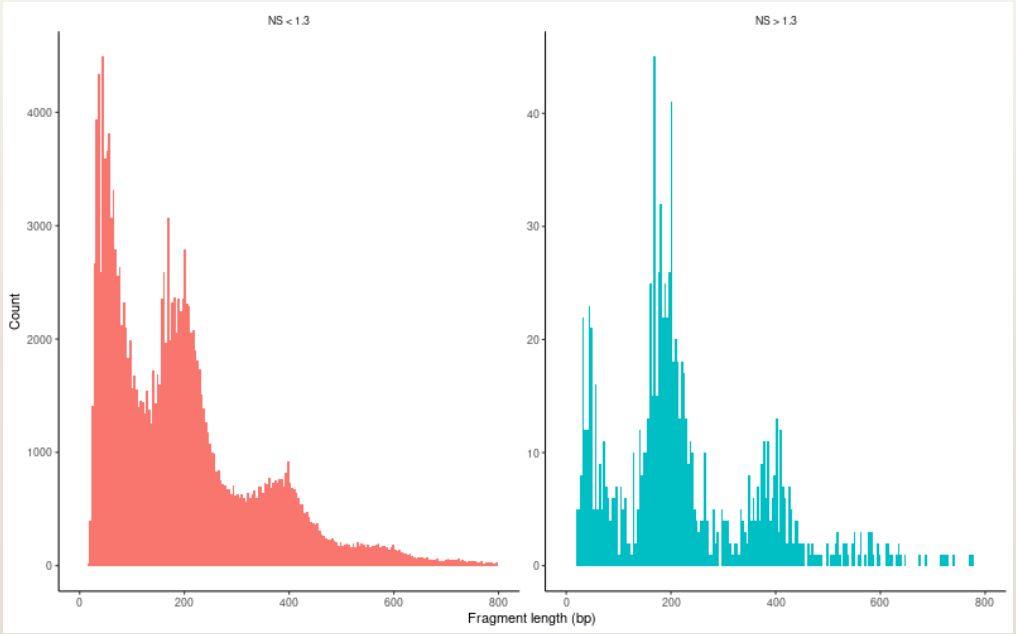


VlnPlot_FeatureScatter.png

解析に使用する細胞のフィルタリング

- PeriodPlotの描画。すべての細胞のフラグメント長の周期性を確認し、ヌクレオソームの信号強度が高いまたは低いセルごとにグループ化することができます。Nucleosome Signalの外れ値にある細胞は、異なるヌクレオソームバンドパターンを持っていることがわかります。

```
> lung$nucleosome_group <- ifelse(lung$nucleosome_signal > 1.3, 'NS > 1.3', 'NS < 1.3')
> png("PeriodPlot.png", width = 800, height = 500)
> FragmentHistogram(object = lung, group.by = 'nucleosome_group', region = 'chr1-1-10000000')
> dev.off()
```

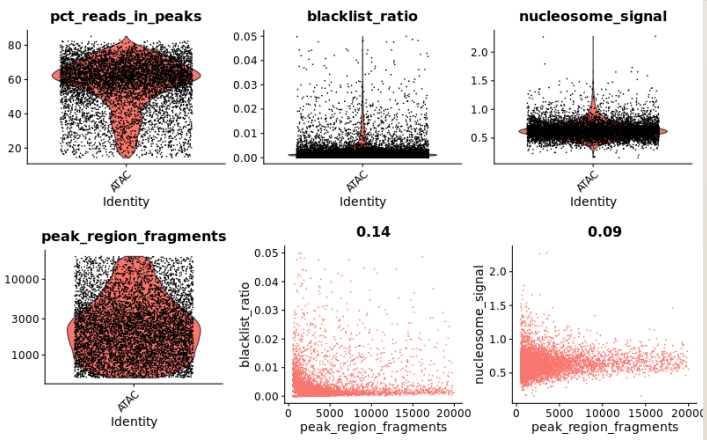


PeriodPlot.png

解析に使用する細胞のフィルタリング

- データ精度の良くない細胞を取り除きます

```
> length(lung@meta.data$nFeature_peaks) #細胞数確認
[1] 8552
>
> #細胞フィルタリング
> lung <- subset(lung, subset = peak_region_fragments > 500 & peak_region_fragments < 20000 &
pct_reads_in_peaks > 10 & blacklist_ratio < 0.05 & nucleosome_signal < 10)
> length(lung@meta.data$nFeature_peaks) #細胞数確認
[1] 7086
>
> #先ほどと同じようにQCの図を描いてみる
> png("VlnPlot_FeatureScatter2.png", width = 800, height = 500)
> plot1 <- VlnPlot(object = lung, features = c('pct_reads_in_peaks', 'blacklist_ratio',
'nucleosome_signal'), pt.size = 0.1) + NoLegend()
> plot2_a <- VlnPlot(object = lung, features = 'peak_region_fragments', pt.size = 0.1, log = TRUE) +
NoLegend()
> plot2_b <- FeatureScatter(lung,"peak_region_fragments",'blacklist_ratio', pt.size = 0.1) + NoLegend()
> plot2_c <- FeatureScatter(lung,"peak_region_fragments",'nucleosome_signal', pt.size = 0.1) + NoLegend()
> plot2 <- plot2_a | plot2_b | plot2_c
> plot1 / plot2
> dev.off()
```

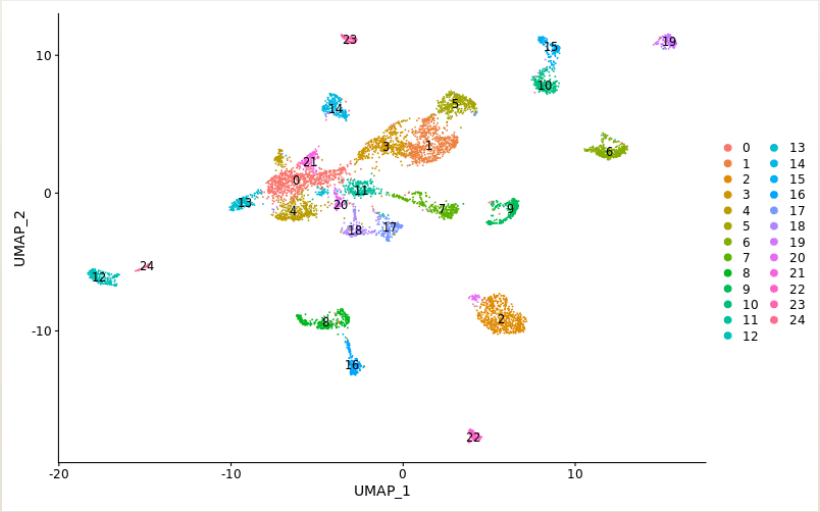


VlnPlot_FeatureScatter2.png
Filtered cells: 500 < peac_region_fragments < 20000,
pct_reads_in_peaks > 10,
blacklist_ratio < 0.05,
nucleosome_signal < 10

クラスタリングとUMAPによるプロット

- フィルターした細胞のピーク情報を用いてクラスタリングとUMAPによる可視化を行います

```
> lung <- RunTFIDF(lung) #term frequency-inverse document frequency (TF-IDF)
normalization
> lung <- FindTopFeatures(lung, min.cutoff = 'q0') #feature selection
>
> lung <- RunSVD(object = lung, assay = 'peaks', reduction.key = 'LSI_',
reduction.name = 'lsi', seed.use= 1234) #dimensional reduction
> lung <- RunUMAP(object = lung, reduction = 'lsi', dims = 1:30)
> lung <- FindNeighbors(object = lung, reduction = 'lsi', dims = 1:30)
> lung <- FindClusters(object = lung, verbose = FALSE)
> png("umap_scATAC.png", width = 800, height = 500)
> DimPlot(object = lung, label = TRUE)
> dev.off()
```



umap_scATAC.png

※ TF-IDFとそれに続くSVDの組み合わせた処理は latent semantic indexing (LSI)として知られています。
https://science.sciencemag.org/content/367/6473/45.ful

アノテーション （細胞種同定）

- scRNA-seqと同様に下記テーブルの既知マーカー遺伝子を用いて細胞種を同定します。

	Cell type		Cell markers
Immune cell Ptprc(CD45)+	T cell (T細胞)		Cd3d, Cd4, Cd8a
	NK cell (NK細胞)		Cd3d(-), Nkg7, Gzma
	B cell (B細胞)		Cd19, Cd79a
	Myeloid cell (骨髄細胞)	Macrophage	Itgam(CD11b)
		Alveolar macrophage	Itgax(CD11c), Siglecf
		Neutrophil (好中球)	Ly6g, Ngp
	DC (樹状細胞)		Itgax(CD11c), Itgae(CD103)
	ILC (Nuocyte or NH) (自然リンパ球)		Cd3d(-), IL2RA(CD25), Gata3+
Non-immune cell	Epithelial cell (上皮細胞)		Epcam, Cdh1
	Myofibroblast (筋繊維芽細胞)		Acta2, Mustn1
	Fibroblast (繊維芽細胞)		Col1a1, Cpl1a2
	Pericyte (周皮細胞)		Mcam, Pdgfrb, Cox4i2
	Endothelial cell (内皮細胞)		Pecam1(CD31), Cdh5, Vwf

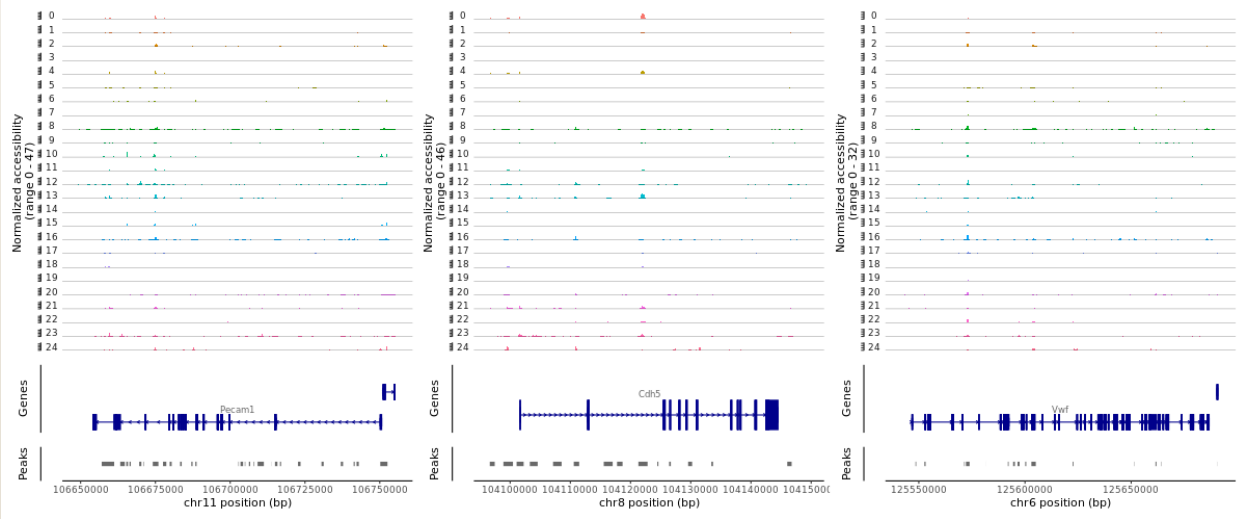
- scRNA-seqのときと同様に、マーカーとなる遺伝子を変数に代入します。

```
> Immune <- c("Ptprc") #CD45
> T <- c("Cd3d", "Cd4", "Cd8a")
> NK <- c("Nkg7", "Gzma")
> B <- c("Cd19", "Cd79a")
> Myeloid <- c("Itgam", "Cd68")
> AlveolarMacrophage <- c("Itgax", "Siglecf", "Cd68")
> DC <- c("Itgax", "Itgae")
> Neutrophil <- c("Ly6g", "Ngp")
> Epithelial <- c("Epcam","Cdh1")
> Myofibroblast <- c("Mustn1","Acta2")
> Fibroblast <- c("Col1a1", "Col1a2")
> Endothelial <- c("Pecam1", "Cdh5", "Vwf")
> Pericyte <- c("Mcam", "Pdgfrb", "Cox4i2")
```

アノテーション （細胞種同定）

- マーカー遺伝子周辺のピーク情報(オープンクロマチンの状態)を見ていきます。

```
> #例として内皮細胞のマーカー遺伝子周辺のピークを確認
> DefaultAssay(lung) <-'peaks'
> gene.coords<-genes(EnsDb.Mmusculus.v79, filter = ~ gene_biotype== "protein_coding")
> seqlevelsStyle(gene.coords) <-'UCSC'
>
> region1 <-GRangesToString(subset(gene.coords, symbol==Endothelial[1]))
> region2 <-GRangesToString(subset(gene.coords, symbol==Endothelial[2]))
> region3 <-GRangesToString(subset(gene.coords, symbol==Endothelial[3]))
>
> png("CoveragePlot_Endothelial.png", width = 1200, height = 500)
> CoveragePlot(object = lung, region = c(region1, region2, region3), extend.upstream=
5000, extend.downstream= 5000, ncol= 3)
> dev.off()
```



CoveragePlot_Endothelial.png

- 同様に symbol==XXXにマーカー遺伝子を入れて画像を作成します。作成した図は https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC/Signac/scATAC_annotation.zip からダウンロードできます。

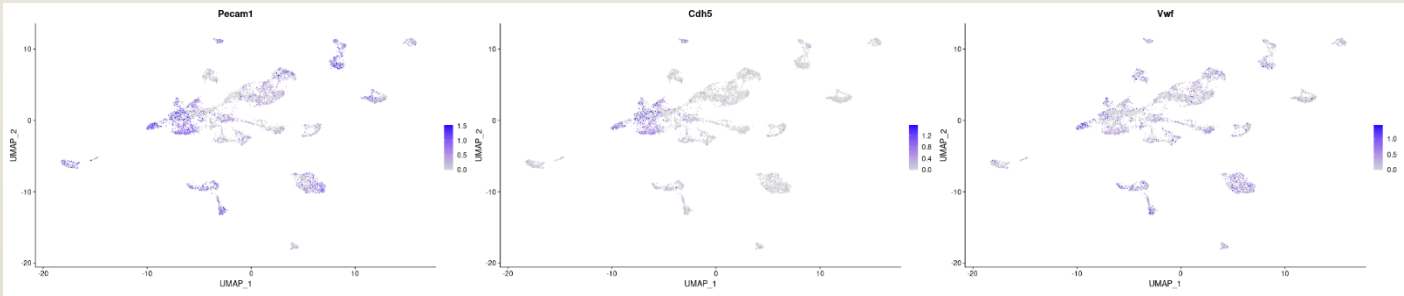
アノテーション （細胞種同定）

- scATAC-seqの結果から“gene activity matrix”(遺伝子とプロモーター領域を含むマトリックス)を作成します。GeneActivity関数によって領域内のフラグメントのカウントまで自動で行われます。

```
> #遺伝子活性マトリックスの作成するために、遺伝子本体+2 kb上流領域の情報を代入
> gene.activities <- GeneActivity(lung)
Extracting gene coordinates
Extracting reads overlapping genomic regions
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=04m 49s
> #scATAC-seqのgene activity matrixをSeuratオブジェクトlungにRNAとして追加後、正規化
> lung[['RNA']] <- CreateAssayObject(counts = gene.activities)
> lung <- NormalizeData(object = lung, assay = 'RNA', normalization.method =
'LogNormalize', scale.factor = median(lung$nCount_RNA))
> DefaultAssay(lung) <- 'RNA'
```

- UMAP上に細胞クラスターをマップしてみます。以下は内皮細胞の例です。

```
> png("FeaturePlot_ATAC_Endothelial.png", width=2400, height=500)
> FeaturePlot(object = lung, features = Endothelial, pt.size = 0.1, max.cutoff = 'q95',
ncol = 3)
> dev.off()
```



- 同様に他の細胞種についても画像を作成します。作成した図は https://kero.hgc.jp/cgi-bin/download/tutorials/learning/analysis_scATAC/Signac/scATAC_annotation.zip からダウンロードできます。

アノテーション （細胞種同定）

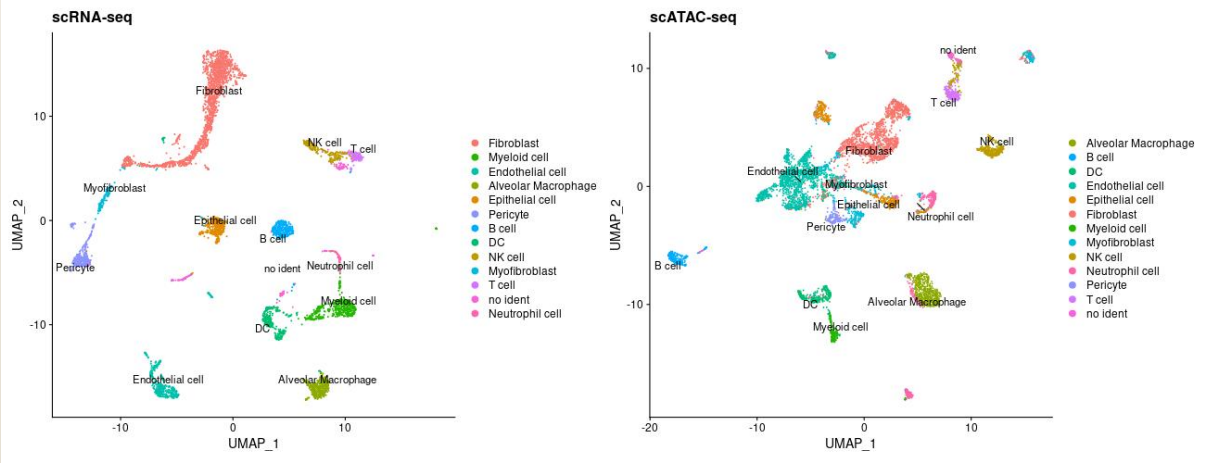
- ここからは、scRNA-seqの結果とscATAC-seqの結果を統合して解析します。
- 前図のようにscATAC-seqの結果から細胞種を人間の目で同定することは、scRNA-seqに比べノイズが多く、かなり困難なことが多いです。
- ここでは、 scRNA-seqのデータを読み込んだ後、FindTransferAnchors (cross-modality integration)とTransferDataという関数を用いて scATAC-seqのデータと対応付けをコンピュータで行います。

```
> #scRNA-seqの解析データ(lung_rnaオブジェクト)を読み込む
> lung_rna <- readRDS("../../analysis_scRNA/Seurat/lung_rna.rds")
> lung_rna$new_cluster <- Idents(lung_rna)
>
> #scRNA-seqとscATAC-seqのクラスターを対応付ける
> #lung_rna: scRNA-seqのSeuratオブジェクト、lung: scATAC-seqのSeuratオブジェクト
> transfer.anchors <- FindTransferAnchors(reference = lung_rna, query = lung,
reduction = 'cca')
> predicted.labels <- TransferData(anchorset = transfer.anchors, refdata =
lung_rna$new_cluster, weight.reduction = lung[['lsi']])
> lung <- AddMetaData(object = lung, metadata = predicted.labels)
```

アノテーション （細胞種同定）

- 上で作成したumap(scATAC-seq)の図に細胞種を同定(マップ)したものを可視化します。

```
> #以下、可視化
> #色がずれるので対応させる
> RNA_Cluster <- unique(lung_rna@active.ident)
> RNA_col <- hue_pal()(length(RNA_Cluster))      #カラーパレットより14色取得
> names(RNA_col) <- RNA_Cluster
> ATAC_Cluster <- unique(predicted.labels$predicted.id)
> ATAC_col <- RNA_col[ATAC_Cluster]
> plot1 <- DimPlot(object = lung_rna, cols = RNA_col, group.by = 'new_cluster', label
= TRUE, repel = TRUE) + ggtitle('scRNA-seq')
> plot2 <- DimPlot(object = lung, cols = ATAC_col, group.by = 'predicted.id', label =
TRUE , repel = TRUE) + ggtitle('scATAC-seq')
>
> png("DimPlot_RNAseq+ATACseq.png", width = 1300, height = 500)
> plot1 + plot2
> dev.off()
```



DimPlot_RNAseq+ATACseq.png

アノテーション （細胞種同定）

- 最後にscRNA-seqとscATAC-seq結果データの対応表を出力してみます。

```
> #scRNA-seqとscATAC-seqデータを統合したときのアノテーションの割当表を出力
> txt <- table(lung$seurat_clusters, lung$predicted.id)
> write.table(txt, file= "table.txt", quote=F, col.names=TRUE, sep="¥t")
```

ATAC cluster	Alveolar Macro-phage	B cell	DC	Endothelial cell	Epithelial cell	Fibroblast	Myeloid cell	Myofibro blast	NK cell	Neutrophil cell	Pericyte	T cell	no ident
2	546	0	0	1	0	0	0	0	0	63	0	0	0
12	0	221	0	0	0	0	0	0	0	0	0	0	0
24	0	30	0	1	0	0	0	0	0	0	0	0	9
8	0	0	276	7	0	2	0	2	2	7	0	0	0
0	0	0	0	679	0	2	0	57	1	2	0	1	0
4	1	0	4	402	2	21	0	3	1	0	1	0	1
13	2	1	1	207	0	4	0	2	1	0	0	0	0
21	0	0	0	126	0	1	0	0	0	0	1	0	0
11	1	1	6	133	0	60	1	12	2	3	1	2	8
23	0	0	6	68	1	1	0	0	2	18	0	0	0
20	32	0	7	53	3	19	3	4	2	6	0	4	4
7	1	0	1	26	225	18	0	37	0	21	2	0	1
14	0	1	1	2	198	3	0	6	1	4	0	0	0
1	0	0	1	9	0	644	0	0	3	3	0	0	0
3	0	0	0	0	0	415	0	30	1	1	1	0	0
5	0	0	0	2	0	368	0	3	2	0	0	0	0
16	0	0	10	1	0	0	184	0	0	0	0	0	0
17	0	0	0	0	0	7	0	179	0	1	0	0	0
19	0	2	1	26	0	19	1	73	1	13	1	0	0
6	0	0	0	0	0	0	0	0	363	0	0	0	0
9	1	0	3	7	48	7	1	10	2	199	0	0	0
22	0	0	0	0	0	0	9	0	0	110	0	0	0
18	0	1	1	0	0	1	0	14	0	1	147	0	0
10	0	0	0	0	0	0	0	0	31	0	0	220	0
15	0	0	0	4	0	0	0	1	87	2	0	1	111

おわり

```
> #lungのオブジェクトデータを保存しておきたい場合
> save (lung, file="lung.Rdata")
> q()
```