

解析環境の構築

1. 以前は...

- バイナリをどこからかダウンロード (簡単)
- ソースコードをどこからかダウンロードして、適当なディレクトリに展開してインストール (makeコマンド) (難易度高い)
- 「PATHを通す」
 1. プログラムをフルパスで指定する (複数プログラムのラッパーの場合うまくいかなかった)
 2. 毎回export PATH=をやる (忘れるとプログラムを探せなくて動かん)
 3. bashrcやbash_profileにかいておく (そもそもbashrcってなに??? という場合どうする)

2. パッケージ管理ツールやコンテナの利用

- パッケージ管理ツール: Anaconda (miniconda, miniforge), Homebrew (Linuxbrew)
- コンテナ(コンテナ型仮想化): Docker, Singularity (Apptainer)

パッケージ管理ツールとは、インストール、バージョン管理、依存関係 (ツール相互の利用) を解決するもの

コンテナとは、OSからツールから全部入りでまるごとそのツールを実行するための環境を提供するもの

Singularityは大元のプロジェクト名がApptainerに変わったので、近い将来遺伝研スパコンでも変更されるかも？

2-1.今回利用するもの

- Miniconda
- Singularity (Apptainer)

MinicondaはAnacondaの簡易版、Miniforge, MambaforgeでもOK

Singularity (Apptainer) はスパコン版のコンテナ

2-2.利点

- makeなどのインストールコマンドを叩かなくてもいい
- パッケージ管理ツールの場合、依存パッケージも一緒に入る
- コンテナの場合、コンテナの中に必要なものがすべて入っている
- バージョン管理が楽
- 環境の移動や共有が楽

2-3.欠点

- 動かないパッケージもある
- (コンテナの場合) ブラックボックスになってしまいがち
- (パッケージ管理ツールの場合) いろいろ入れているうちにcondaそのものが壊れてしまうこともある

3. RNA-seq解析環境

この後の解析実習では以下のツールを使う

- HISAT2
- StringTie
- Samtools

これらを、Minicondaの仮想環境にインストールする方法と、Singularityコンテナを利用する方法を紹介

3-1. Miniconda(Miniforge)

Minicondaまたはminiforge (Anaconda) を既にインストールしている場合は **3-1-2. 仮想環境の作成** から始める (conda-forgeを最優先channelにする)

3-1-1. Miniconda (miniforge) のインストール

Miniforgeをインストールする手順;

- miniforgeはパッケージレポジトリ (channel) のデフォルトがconda-forge

miniforge

<https://github.com/conda-forge/miniforge>

1. Miniforgeのインストールスクリプト

GitHubのDownload <https://github.com/conda-forge/miniforge#download> から Linux用のインストールスクリプト **Miniforge3-Linux-x86_64.sh** を入手

2. スパコンにログイン

```
qlogin
```

3. 自分のホームディレクトリにツールをインストールするディレクトリを作成

(デモでは**tools**というディレクトリ名を使用する)

```
mkdir tools
```

4. ツール用ディレクトリに移動

```
cd tools
```

5. インストールスクリプトをスパコンにコピー

wgetで直接入手

GitHubのInstall <https://github.com/conda-forge/miniforge#install> に従う

```
curl -L -O "https://github.com/conda-  
forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$(uname -  
m).sh"
```

または

```
wget "https://github.com/conda-  
forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$(uname -  
m).sh"
```

6. 画面の説明通りにインストール

```
bash Miniforge3-Linux-x86_64.sh
```

実行後、ライセンスなどの表示が出るので、enterでスクロールして画面の指示に従う

```
Do you accept the license terms? [yes|no]  
[no] >>> yes
```

デフォルトだとアカウントのホームディレクトリ直下にインストールされる

/home/アカウント名/miniconda3

今回 **tools** ディレクトリにインストールしたいので、以下のように書き換える

/home/アカウント名/tools/miniforge3

コピペだとうまくいかないことがあるので、注意深く手打ちで入力する

```
Miniconda3 will now be installed into this location:  
/home/アカウント名/miniconda3
```

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

```
[/home/アカウント名/miniforge3] >>> /home/アカウント名/tools/miniforge3
```

個別パッケージのインストールが終了すると `conda init` を実行するかどうか聞かれるので `yes` と入力

WARNING:

```
You currently have a PYTHONPATH environment variable set. This may
cause
unexpected behavior when running the Python interpreter in Miniforge3.
For best results, please verify that your PYTHONPATH only points to
directories of packages that are compatible with the Python
interpreter
in Miniforge3: /home/koshu2/tools/miniforge3
Do you wish the installer to initialize Miniforge3
by running conda init? [yes|no]
[no] >>> yes
```

完了すると以下のように表示された

```
==> For changes to take effect, close and re-open your current shell. <==

If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniforge3!
```

参考URL

<https://github.com/conda-forge/miniforge#unix-like-platforms-mac-os--linux>

7. condaの反映

インタラクティブノードから一旦出て、もういちどqlogin

```
exit
qlogin
```

プロンプトに

```
(base) [アカウント名@ノード名 ~]$
```

と表示されていることを確認

nodeはいまログインしているノード (`at137`など)、directoryは今いるところ (qloginしたばかりなので `~`)

8. condaの初期設定

- 入っているパッケージを確認(インストールができていないかの確認でもある)

```
conda list
```

- channelsの優先度の確認

```
conda config --get channels
```

以下のように出力されるはず (conda-forgeが優先される)

```
--add channels 'defaults' # lowest priority  
--add channels 'conda-forge' # highest priority
```

- その他設定 (今回は行いません)

condaのupdate

```
conda update -n base conda
```

baseが表示されるのが嫌な時 <https://ja.stackoverflow.com/questions/61630/anacondaでbaseが自動的にactivateされるのを防ぎたい> などを参考に `~/.condarc` で設定する

3-1-2. 仮想環境の作成

ツール間の依存関係があるため、あるツールをインストールするとほかのツールが動かなくなることがある

それを防ぐために仮想環境を作成し、そのなかにツールをインストールする

1. 仮想環境をつくる

toolsディレクトリに移動

```
cd tools
```

pags_rnaseqという名で仮想環境作成

```
conda create -n pags_rnaseq
```

Proceed ([y]/n)? と聞かれるので **y** と入力

```
## Package Plan ##

  environment location: /home/xxxxx/tools/miniforge3/envs/pags_rnaseq

Proceed ([y]/n)? y
```

完了すると以下のように表示される

```
#
# To activate this environment, use
#
#     $ conda activate pags_rnaseq
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

2. 仮想環境の有効化

```
conda activate pags_rnaseq
```

プロンプトに

```
(pags_rnaseq)[アカウント名@ノード名 tools] $
```

と表示されていることを確認

3. 仮想環境の終了

```
conda deactivate
```

プロンプトに

```
(base)[アカウント名@ノード名 tools] $
```

と表示されていることを確認

3-1-3. 解析ツールのインストール

次の実習で使うツールのバージョン

- HISAT2 v2.2.1
- StringTie v2.2.1
- Samtools v1.17

1. 仮想環境 pags_rnaseq に入る

```
conda activate pags_rnaseq
```

プロンプトに

```
(base)[アカウント名@ノード名 tools] $
```

と表示されていることを確認

2. installコマンドを調べる

`conda hisat2`, `conda stringtie`, `conda samtools`で検索 -> Anaconda.orgのページでコマンド確認

複数のパッケージが出てくるので、biocondaチャンネルのパッケージを選ぶ

3. HISAT2のインストール

condaより高速なmambaコマンドを使用

参考 <https://kazumaxneo.hatenablog.com/entry/2021/02/11/073000>

パッケージ名=バージョンを付けてバージョン指定でインストール

```
mamba install -c bioconda hisat2=2.2.1
```

途中でインストールの可否について聞かれるので、問題なければ **Y** と入力する

インストールができていることを確認

```
hisat2 -h
```

ヘルプメッセージが出たらOK

4. StringTieのインストール

```
mamba install -c bioconda stringtie=2.2.1 -y
```

-y をつけると途中でインストールの可否(yes/no)を聞かれずにそのまま進む

インストールができていることを確認

```
stringtie -h
```

ヘルプメッセージが出たらOK

5. Samtoolsのインストール

```
mamba install -c bioconda samtools=1.17 -y
```

インストールができていることを確認(**-h**は不要)

```
samtools
```

ヘルプメッセージが出たらOK

6. パッケージリストの作成

```
conda env export -n pags_rnaseq > pags_rnaseq.yml
```


`pags_rnaseq.yml`を使うと、同じ環境(依存パッケージの構成やバージョン)が作成できる(今回は行いません)

次に進む前に、いったん仮想環境から出る

```
conda deactivate
```

プロンプトに

```
(base) [アカウント名@ログインしているノード tools] $
```

と表示されていることを確認

3-2. コンテナの利用

- プロジェクト名はApptainerに変わったが、2023年10月現在、遺伝研スパコンではまだsingularityコマンドが使える

3-2-1. コンテナの場所

`/usr/local/biotools/ツールの頭文字`以下にコンテナがある 詳しい説明は遺伝研スパコンのhomepageを見る

HISAT2

```
ls /usr/local/biotools/h/hisat2*
```

StringTie

```
ls /usr/local/biotools/s/stringtie*
```

Samtools

```
ls /usr/local/biotools/s/samtools*
```

バージョンたくさんあるが、一番新しいものを使う(動かないものもあるので、そのときは別のバージョンを使う)

3-2-2. コンテナを動かす

`singularity exec オプション コンテナのフルパス ツールのコマンド ツールのオプション`

というかたちで動かす

以下を1行で入力、実行

```
singularity exec /usr/local/biotools/s/stringtie:2.2.1--hecb563c_2  
stringtie -h
```

実行すると以下のヘルプメッセージが出力される

```
StringTie v2.2.1 usage:  
  
stringtie <in.bam ..> [-G <guide_gff>] [-l <prefix>] [-o <out.gtf>] [-p  
<cpus>]  
  [-v] [-a <min_anchor_len>] [-m <min_len>] [-j <min_anchor_cov>] [-f  
<min_iso>]
```

```
[-c <min_bundle_cov>] [-g <bdist>] [-u] [-L] [-e] [--viral] [-E  
<err_margin>]  
[--ptf <f_tab>] [-x <seqid,..>] [-A <gene_abund.out>] [-h] {-B|-b  
<dir_path>}  
[--mix] [--conservative] [--rf] [--fr]  
Assemble RNA-Seq alignments into potential transcripts.  
(以下略)
```

自分のhomeディレクトリ以外のファイルを読み込みたい時は、**-B** オプションでbindする