

# 解析環境の構築

国立遺伝学研究所 大量遺伝情報研究室 坂本美佳

# 初心者の壁

インストール

# 初心者の壁

## 解析ツールのインストール

- ・バイナリファイルをどこかに置く（簡単）
- ・ソースコードをコンパイルしてインストール（難易度 高）



# 初心者の壁

## 解析ツールのインストール

- PATHを通す
  1. プログラムをフルパスで指定  
複数プログラムのラッパーの場合うまくうごくかわからん
  2. 每回 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に書いておく  
そもそもbashrcってなに？？？という場合どうする

# 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
- ・ コンテナ(コンテナ型仮想化)



# 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)**  
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
- ・ コンテナ  
**Docker, Apptainer(Singularity)**  
まるごとそのツールを実行するための  
「環境(OS含む）」を提供  
コンピュータの中でもうひとつ別のコン  
ピュータを動かすイメージ

# 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

**Miniconda (Miniforge)**

Anacondaの簡易版

- ・ コンテナ

**Apptainer (Singularity)**

スパコン版のコンテナ

# パッケージ管理ツールやコンテナの利用

- 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) 依存パッケージも一緒にに入る
  - ✓ (コンテナ) 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

# パッケージ管理ツールやコンテナの利用

- ・ 欠点
  - ✓ 動かないパッケージ...
  - ✓ (パッケージ管理ツール) いろいろ入れているうちに管理ツール 자체が壊れてしまうことも
  - ✓ (コンテナ) ブラックボックスになりがち

とはいえ、利点がより多いので積極的に使っていきましょう

# RNA-seq解析環境

## このあと利用するツール

- HISAT2 リードをリファレンス配列にマッピング
- StringTie 転写産物isoformの同定
- Samtools sam/bamファイルの操作

### 1. conda仮想環境にインストールする方法

### 2. コンテナを利用する方法

次はMiniforgeの利用

# Miniforgeの利用

## 概要

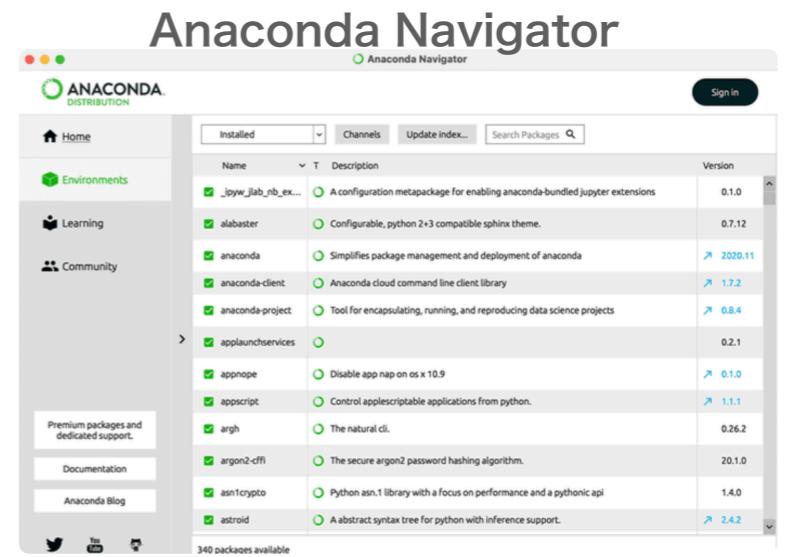
- Miniforgeのインストールと初期設定
- 仮想環境の作成 <- condaインストール済みのひとはここから
- 仮想環境に解析ツールをインストール

コマンドは手打ちorハンドアウトからコピペでOK

# Miniforgeの利用

## Anaconda/Miniconda/Miniforge

- ✓ Pythonのディストリビューション
- ✓ パッケージ管理ツールcondaを含む
- ✓ AnacondaはGUIツールを含む多くのパッケージ
- ✓ Miniconda(Miniforge)は最低限必要なパッケージのみ
- ✓ Miniforgeはmambaが初めから含まれている (2023.9~)



今回は、conda-forgeからパッケージをインストールする  
Miniforgeインストーラを利用します

参考

<https://tooljp.com/windows/chigai/html/Python/conda-anaconda-miniconda-pyenv-pip-chigai.html>

# Miniforgeの利用

## Miniforgeとは

- Anaconda/Miniconda
  - ✓ 個人、公認教育機関の学生・教員、オープンソース開発者、中小企業などは無料（それ以外は有償）
- conda-forge
  - ✓ コミュニティベースで開発
  - ✓ オープンソース

次は仮想環境の作成

# Miniforgeの利用

## Miniforgeのインストール

コマンド

出力を確認

入力が必要なところは**文字色を変えて**示します

# 遺伝研スパコンアカウントがある場合

アカウントがない人は、自分のPCの環境に合わせてやってみてください  
今回は、自分のPCへのインストールの方法は扱いません

# Miniforgeの利用

## Miniforgeインストーラ

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

The screenshot shows the GitHub repository page for `miniforge`. The repository has 5 branches and 79 tags. The main branch has 602 commits. Recent commits include:

- Bump actions/checkout from 3 to 4 (3 weeks ago)
- use cls 23.3.0 (last month)
- restore deeper docs path (3 years ago)
- Always check for boa compatibility (last month)
- consolidate scripts (4 years ago)
- Update copyright date. (last year)
- Merge pull request #488 from hmaarrfk/recommend\_miniforge (11 hours ago)
- Remove extra hack (last year)
- Update micromamba for osx (last year)
- Fix shellcheck findings (2 years ago)

The `README.md` file contains the following text:

This repository holds a minimal installer for [Conda](#) specific to [conda-forge](#). Miniforge allows you to install the conda package manager with the following features pre-configured:

- [conda-forge](#) set as the default (and only) channel.
  - Packages in the base environment are obtained from the [conda-forge channel](#).
- Optional support for PyPy in place of standard Python interpreter (aka "CPython").
- Optional support for [Mamba](#) in place of Conda.
- An emphasis on supporting various CPU architectures (x86\_64, ppc64le, and aarch64 including Apple M1).

# Miniforgeの利用

## Miniforgeインストーラ

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

☰ README.md

It can be compared to the [Miniconda](#) installer.

### Download ↗

Miniforge installers are available here: <https://github.com/conda-forge/miniforge/releases>

#### Miniforge3 ↗

Latest installers with Python 3.10 (\*) in the base environment:

OS	Architecture	Download
Linux	x86_64 (amd64)	<a href="#">Miniforge3-Linux-x86_64</a>
Linux	aarch64 (arm64) (**)	<a href="#">Miniforge3-Linux-aarch64</a>
Linux	ppc64le (POWER8/9)	<a href="#">Miniforge3-Linux-ppc64le</a>
OS X	x86_64	<a href="#">Miniforge3-MacOSX-x86_64</a>
OS X	arm64 (Apple Silicon) (***)	<a href="#">Miniforge3-MacOSX-arm64</a>
Windows	x86_64	<a href="#">Miniforge3-Windows-x86_64</a>

(\*) The Python version is specific only to the base environment. Conda can create new environments with different Python versions and implementations.

(\*\*) For Raspberry Pi that include a 64 bit processor, you must also use a 64-bit operating system such as [Raspberry Pi OS 64-bit](#) or [Ubuntu for Raspberry Pi](#). The versions listed as "System: 32-bit" are not compatible with the installers on this website.

(\*\*\*) Apple silicon builds are experimental and haven't had testing like the other platforms.

このリンクをコピーして  
wgetまたはcurlでダウンロード

# Miniforgeの利用

## Miniforgeのインストール

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

The screenshot shows the GitHub page for the Miniforge repository. The README.md file is displayed, which includes a comparison to the Miniconda installer and a 'Download' section. The 'Install' section is highlighted with a red border. It provides instructions for Unix-like platforms (Mac OS & Linux) using curl or wget, followed by command-line scripts for both methods. A note at the bottom states that Raspberry Pi users must use a 64-bit operating system.

☰ README.md

It can be compared to the [Miniconda](#) installer.

[Download](#)

☰ README.md

[Install](#)

**Unix-like platforms (Mac OS & Linux)**

Download the installer using curl or wget or your favorite program and run the script. For eg:

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

or

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

(\*\*) For Raspberry Pi that include a 64 bit processor, you must also use a 64-bit operating system such as [Raspberry Pi OS 64-bit](#) or [Ubuntu for Raspberry Pi](#). The versions listed as "System: 32-bit" are not compatible with the installers on this website.

(\*\*\*) Apple silicon builds are experimental and haven't had testing like the other platforms.

# Miniforgeの利用

## Miniforgeのインストール

スパコンにログイン+ `qlogin`

ホームディレクトリにツール用のディレクトリを作成

```
mkdir tools
```

ツール用ディレクトリに移動

```
cd tools
```

`wget` コマンドでインストールスクリプトをダウンロード

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

1行で (改行しないで) 入力してください

GitHubからコマンドをコピーすると楽ですが、いったんテキストエディタにペーストして確認しましょう

# Miniforgeの利用

## Miniforgeインストーラ FileZilla利用

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

☰ README.md

It can be compared to the [Miniconda](#) installer.

**Download** ↗

Miniforge installers are available here: <https://github.com/conda-forge/miniforge/releases>

**Miniforge3** ↗

Latest installers with Python 3.10 (\*) in the base environment:

OS	Architecture	Download
Linux	x86_64 (amd64)	<a href="#">Miniforge3-Linux-x86_64</a>
Linux	aarch64 (arm64) (**)	<a href="#">Miniforge3-Linux-aarch64</a>
Linux	ppc64le (POWER8/9)	<a href="#">Miniforge3-Linux-ppc64le</a>
OS X	x86_64	<a href="#">Miniforge3-MacOSX-x86_64</a>
OS X	arm64 (Apple Silicon) (***)	<a href="#">Miniforge3-MacOSX-arm64</a>
Windows	x86_64	<a href="#">Miniforge3-Windows-x86_64</a>

(\*) The Python version is specific only to the base environment. Conda can create new environments with different Python versions and implementations.

(\*\*) For Raspberry Pi that include a 64 bit processor, you must also use a 64-bit operating system such as [Raspberry Pi OS 64-bit](#) or [Ubuntu for Raspberry Pi](#). The versions listed as "System: 32-bit" are not compatible with the installers on this website.

(\*\*\*) Apple silicon builds are experimental and haven't had testing like the other platforms.

このリンクからダウンロード

# Miniforgeの利用

## Minicondaのインストール FileZilla利用

スパコンにログイン+ qlogin、ホームディレクトリにツール用のディレクトリを作成

```
mkdir tools
```

FileZillaを用いてインストールスクリプト **Miniforge3-Linux-x86\_64.sh** を  
**tools**ディレクトリにコピー

ツール用ディレクトリに移動

```
cd tools
```

# Miniforgeの利用

## Miniforgeのインストール

インストールスクリプトの実行

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

スクロールして次の文章が表示されたら **yes** と入力

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

インストールディレクトリの変更 何もしないと/home/アカウント名/miniforge3にインストールされます

```
Miniconda3 will now be installed into this location:  
/home/アカウント名/miniforge3
```

- 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
```

# Miniforgeの利用

## Miniforgeのインストール

必要最低限パッケージのインストールが終わると `conda init` を実行するか聞かれる  
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 Miniconda3.  
For best results, please verify that your PYTHONPATH only points to directories of packages that are compatible with the Python interpreter in Miniconda3: /home/アカウント名/tools/forge3

Do you wish the installer to initialize Miniforge3  
by running `conda init`? [yes|no]  
[no] >>> yes

~/.bashrcに設定が書き込まれます

いったん`exit`し、ふたたび`qlogin`

プロンプトに(`base`)と表示されていることを確認

(`base`) [アカウント名@ノード名 ~]\$

次は初期設定

# Miniforgeの利用

## 初期設定

パッケージリストの表示

```
conda list
```

チャンネルの優先度の確認

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

以下のように出力されるはず

```
--add channels 'defaults' # lowest priority
```

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

Anaconda/Minicondaが既にインストールされていた場合

conda-forge チャンネルの追加 パッケージの取得元をconda-forge優先にします

```
conda config --add channels conda-forge
```

次は仮想環境

# Miniforgeの利用

## 仮想環境の作成

- ・ ツール（ソフトウェア）間に依存関係があるため、
  - ✓ あるツールを動かすためにほかのツールが必要
  - ✓ あるツールをインストールすると他のツールが動かなくなることもある

これを防ぐために

- 仮想環境を作成し、その中にツールをインストール
- うまく動かなかったら仮想環境を破棄して作り直すことは簡単

# Miniforgeの利用

## 仮想環境の作成

toolsディレクトリに移動

ここまで流れではhomeディレクトリにいるはず

```
cd tools
```

pags\_rnaseqという名の仮想環境を作成

```
conda create -n pags_rnaseq
```

Proceed ([y]/n)? と聞かれたら **y** と入力

```
## Package Plan ##
```

```
environment location: /home/xxxxx/tools/miniconda3/envs/pags_rnaseq
```

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

# Miniforgeの利用

## 仮想環境の作成

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    仮想環境に入るコマンド  
#  
# To deactivate an active environment, use  
#  
#     $ conda deactivate    仮想環境から出るコマンド
```

# Miniforgeの利用

## 仮想環境の作成

仮想環境の有効化

```
conda activate pags_rnaseq
```

プロンプトが以下のように表示されていることを確認

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

↑ 仮想環境名

仮想環境の終了

```
conda deactivate
```

プロンプトが以下のように表示されていることを確認

```
(base) [アカウント名@ノード名 tools] $
```

↑ baseに戻る

次は解析ツールのインストール

# Miniforgeの利用

## 解析ツールのインストール

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

# Miniforgeの利用

## 解析ツールのインストール

- インストールコマンドの調べ方



# Miniforgeの利用 解析ツールのインストール

The screenshot shows the Anaconda.org website interface. At the top, there is a navigation bar with the Anaconda logo, a search bar, and links for About, Anaconda, Help, Download Anaconda, Sign In, and Account. Below the header, the main content area displays the package details for 'hisat2' version 2.2.1. The title 'bioconda / packages / hisat2 2.2.1' is prominently displayed. To the right of the title is a green navigation bar with icons for Home, Development, Documentation, Downloads, and a star icon. Below the title, a brief description states: 'Graph-based alignment of next generation sequencing reads to a population of genomes.' A navigation bar below the title includes tabs for Conda (which is selected), Files, Labels, and Badges. The 'Conda' tab section contains links to License (GPL-3.0), Home (http://daehwankimlab.github.io/hisat2), Development (https://github.com/DaehwanKimLab/hisat2), Documentation (https://daehwankimlab.github.io/hisat2/manual/), Total downloads (260087), and Last upload (4 months and 9 days ago). A large 'Installers' section follows, featuring a note about non-standard labels and two download options: 'linux-64 v2.2.1' and 'osx-64 v2.2.1'. A large red box highlights the 'linux-64' link. Below this, a 'conda install' command is shown: 'conda install -c bioconda hisat2'. A second 'conda install' command is also present: 'conda install -c "bioconda/label/cf201901" hisat2'. A red box highlights the entire 'conda install' command. To the right of the command, the text 'インストールコマンド' (Installation Command) is written in red.

bioconda / packages / hisat2 2.2.1

Graph-based alignment of next generation sequencing reads to a population of genomes.

Conda Files Labels Badges

License: GPL-3.0  
Home: <http://daehwankimlab.github.io/hisat2>  
Development: <https://github.com/DaehwanKimLab/hisat2>  
Documentation: <https://daehwankimlab.github.io/hisat2/manual/>  
260087 total downloads  
Last upload: 4 months and 9 days ago

Installers

Info: This package contains files in non-standard labels.

linux-64 v2.2.1 最新バージョン  
osx-64 v2.2.1

conda install ?

To install this package run one of the following:

```
conda install -c bioconda hisat2
conda install -c "bioconda/label/cf201901" hisat2
```

インストールコマンド

# Miniforgeの利用

## 解析ツールのインストール

HISAT2のインストール

仮想環境の有効化

```
conda activate pags_rnaseq
```

パッケージ名=バージョンとするとバージョンを指定してインストールできる

高速なmambaコマンドを利用

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

インストールの可否について聞かれるので **y** と入力

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

ヘルプメッセージの表示（インストールの確認）

```
hisat2 -h
```

# Miniforgeの利用

## 解析ツールのインストール

StringTieのインストール **-y** を付けると、インストールの可否について聞かれずに進む

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

ヘルプメッセージの表示（インストールの確認）

```
stringtie -h
```

Samtoolsのインストール

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

ヘルプメッセージの表示（インストールの確認）

```
samtools
```

# Miniforgeの利用

## パッケージリストの作成

仮想環境 `pags_rnaseq` のパッケージリスト

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

`pags_rnaseq.yml` を使うと同じ環境（依存パッケージやバージョン）が作成可能

今回は行いません

仮想環境から出る

```
conda deactivate
```

プロンプトが以下のように表示されていることを確認

```
(base) [アカウント名@ノード名 tools] $
```

次はSingularityコンテナの利用

# Apptainer(Singularity)の利用 概要

- ・ コンテナの探し方
- ・ コンテナの動かし方

# Apptainer(Singularity)の利用

## コンテナの探し方

コンテナの場所 `/usr/local/biotools/ツールの頭文字`

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

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

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

- ・一番新しいバージョンを使う
- ・(ただし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)

# Apptainer(Singularity)の利用

## コンテナの動かし方

### 基本

`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`

### 練習 (StringTieのヘルプメッセージ表示)

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

ハンドアウトから、または lsコマンドで表示されたパスをコピーしてください

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)

では実際に使ってみましょう

