支持生物序列机
使用手册 V1.0

2023 年 10 月

目　录

第 1 章 前 言

1.1 背景

支持生物序列机是生物信息学领域中生物序列分类任务的软件包。

生物信息学是计算机科学和生物学的交叉学科。在传统的计算机科学中，有多个研究对象，例如自然语言处理针对文本或语音进行研究，物体识别和目标检测针对图像或视频进行研究，而生物信息学的研究对象主要以蛋白质、DNA 和 RNA 为主。目前，支持生物序列机 1.0 版本仅支持蛋白质序列的分类，以 DNA 和 RNA 为分析对象的支持生物序列机仍在研究中。

蛋白质分类问题是生物信息学的蛋白质组学中的经典问题，利用机器学习实现蛋白质分类的研究已经取得了多项成果，并为生物学和医学研究提供了帮助。蛋白质分类问题通过构建各种适合的机器学习算法，在训练集上，以蛋白质序列为输入，以蛋白质类别为输出，来训练模型；在测试集上，通过输入蛋白质序列，获得预测标签，然后选择合适的性能评价指标（例如真正类率、真负类率、精确率等）来评估和优化模型。

1.2 方法简介

支持生物序列机主要使用了多序列比对技术、多核学习和支持向量机，其示意图如图 1-1 所示。

支持生物序列机定义了一个名为 PSD 的标准过程。在蛋白质序列比对的过程中，氨基酸字母表的庞大尺寸导致了过多的间隙插入，破坏了序列的对齐，从而降低了蛋白质序列之间相似性的表示的准确性。PSD 过程通过对氨基酸理化性质进行谱聚类，然后将氨基酸分组，有效降低了氨基酸的字母表，建立了原始蛋白质序列与其结构之间的联系。

支持生物序列机引入了序列比对来度量蛋白质的相似性，生成了蛋白质的相似度核矩阵。支持生物序列机还提出了一个全新的多核学习方法，即混合中心核依赖性最大化多核学习（HCKDM-MKL）。这种方法被创新地引入到序列分类中。生物学中有许多方法可以计算蛋白质之间的相似性，从而生成相应的核矩阵。HCKDM-MKL

的引入，提供了从不同角度整合蛋白质多元信息的可能性，从而使研究者能够通过设计各种序列相似性度量来有效地提高序列分类的准确性。

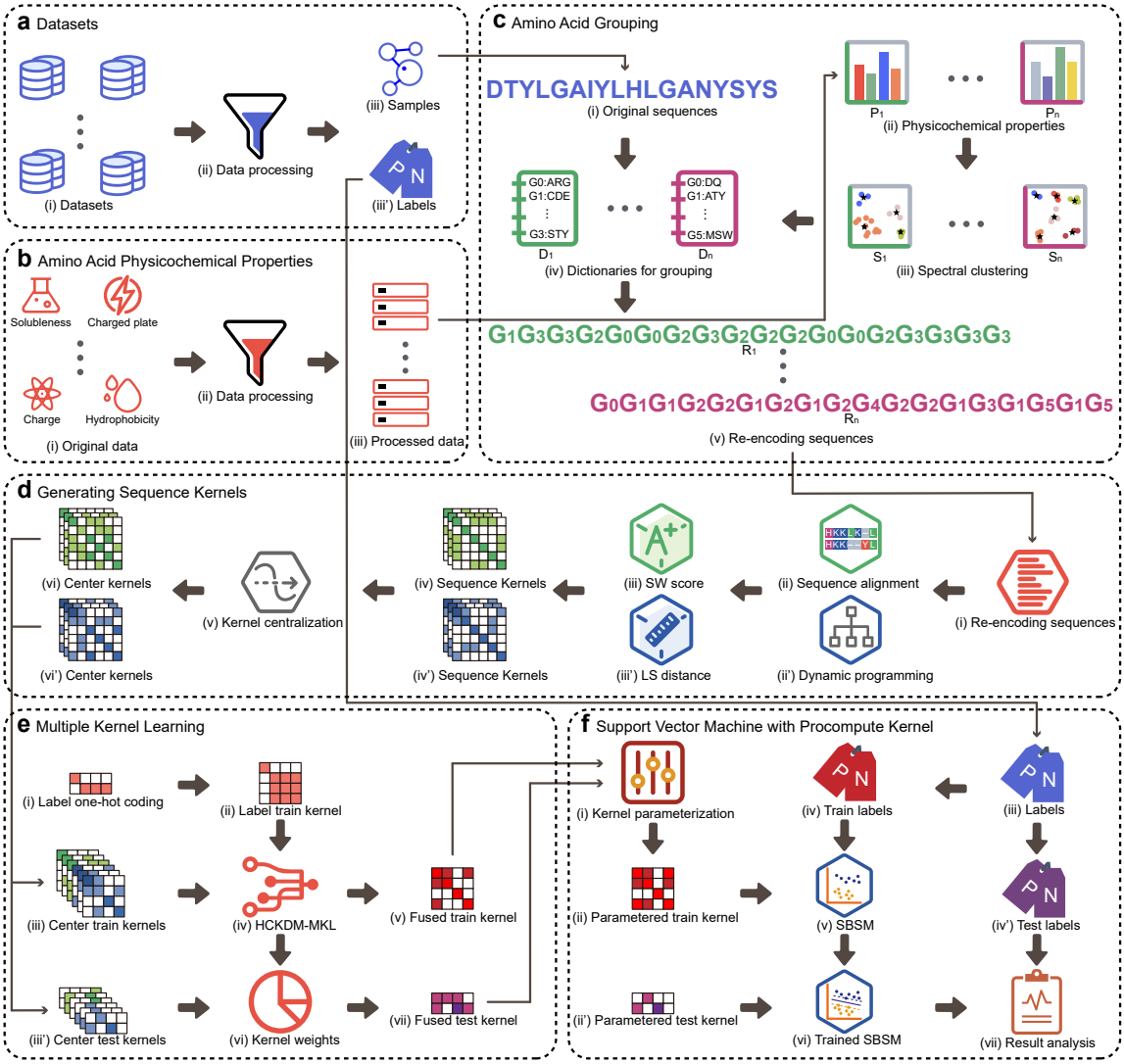


图 1-1 支持生物序列机的流程图

支持生物序列机以支持向量机为基石，将输入到模型的数字向量替换成原始的蛋白质序列，有效地避免了特征信息的丢失，保留了潜在的规律信息和关联模式。与传统方法相比，支持生物序列机在蛋白质功能和修饰位点识别的 10 个数据集中都表现出色。这项研究不仅展示了蛋白质分类领域的最新成果，还成为专门为生物序列构建分类算法框架的一次有益实践。

1.2 技术支持

支持生物序列机得在线网站已经开发上线，网址是 http://lab.malab.cn/soft/SBSM-Pro/。网站内容如图 1-2 所示。该网站会发布新版软件包、发布 bug 修复，更新代码文档和使用手册等，请关注该网站的更新内容，以便获得最新的支持生物序列机。

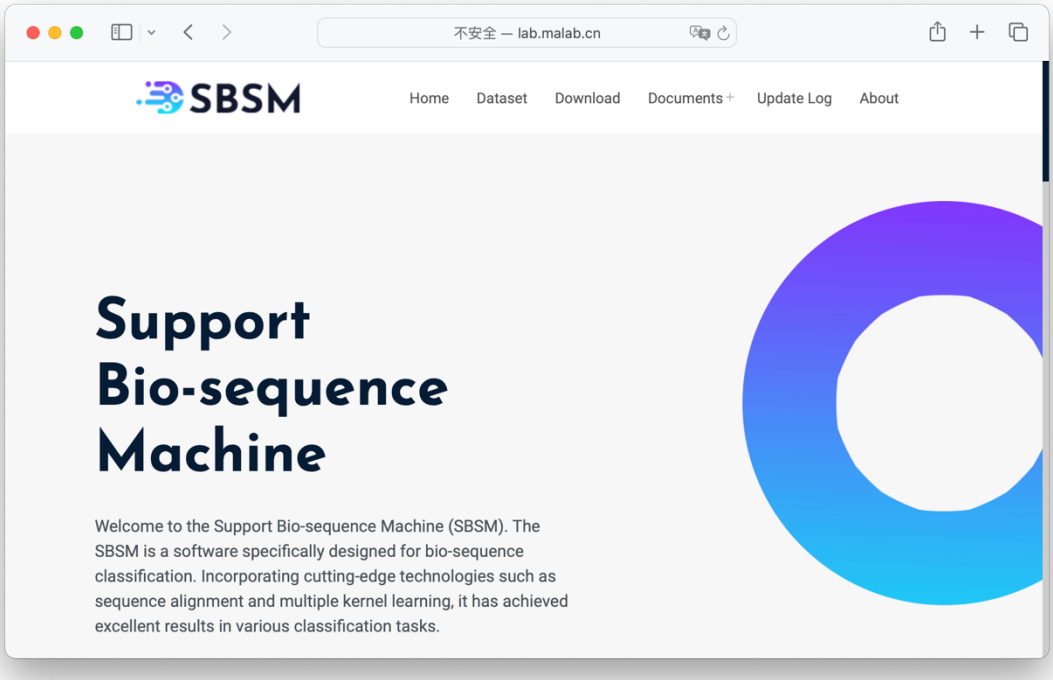


图 1-1 支持生物序列机的在线网站

支持生物序列机的论文已经发布在预印本网站 arXiv 上，网址是 https://arxiv.org/abs/2308.10275，其中详细说明了支持生物序列机的原理和思想，供您参考。

支持生物序列机的维护者是王一争，邮箱地址是 wyz020@126.com，微信号是 biowyz，如果在使用支持生物序列的过程中，您有任何疑问或建议，欢迎您与我们联系，这对我们进一步完善软件很重要，感谢支持！

第 2 章 Python Package 版本

2.1 Python Package

在支持生物序列机的网站中可以获得 Python 的 Package 版本的支持生物序列机网址是 http://lab.malab.cn/soft/SBSM-Pro/Python.html。其使用方法与 numpy、scikit-learn 这种 Python 的 Package 相同，通过 import 导入相关包，然后调用其中方法即可。Python Package 支持 Windows、Linux 多个系统的使用。

2.2 whl 文件下载与安装

在支持生物序列机的网站可以下载 whl 文件，whl 文件是一个 Python Wheel 文件，它是一个用于分发 Python 项目的包格式。Wheel 是一个二进制包格式，它允许预编译的扩展和其依赖项捆绑在一起，使得安装更快速且不需要本地编译。

你可以使用 pip 工具来安装 whl 文件。例如，对于支持生物序列机的某一个版本，它的 wheel 文件名是 sbsm-1.0-py3-none-any.whl，那么使用 pip 安装支持生物序列的命令如下所示：

```
pip install sbsm-1.0-py3-none-any.whl
```

2.3 Python Package 的使用

在成功安装后，您应首先在您的 python 文件中导入 sbsm：

```
import sbsm
```

为了获得更高的运行速度，我们整合了`multiprocessing`模块进行并行化。这要求用户在`if __name__ == "__main__"`下使用保护，并通过使用`freeze_support()`方法避免错误：

```
if __name__ == "__main__":
    from multiprocessing import freeze_support
    freeze_support()
```

接下来，创建一个 sbsm 对象，在这时您可以指定参数。

```
cls = sbsm.SBSM(c=64)
```

使用`fit()`方法训练模型，其中输入是训练样本的原始蛋白质序列及其相应的标签。

```
cls.fit(X_train, y_train)
```

使用`predict()`方法预测，其中输入是预测样本的原始蛋白质序列。

```
y_predict = cls.predict(X_test)
```

下面显示了完整的使用示例：

```
import sbsm
import numpy as np

if __name__ == '__main__':
    from multiprocessing import freeze_support
    freeze_support()

    # 加载你自己的数据
    X_train = read_fasta('train_sample.fasta')  # 训练的原始蛋白质序列
     y_train = read_txt('train_label.txt')  # 训练的相应标签
     X_test = read_fasta('test_sample.fasta')  # 测试的原始蛋白质序列
     # y_test = read_txt('test_label.txt')  # 测试的相应标签

     # 训练和测试 SBSM
     cls = sbsm.SBSM(c=10)  # 定义 SBSM 对象
     cls.fit(X_train, y_train)  # 训练 SBSM
     y_predict = cls.predict(X_test)  # 测试 SBSM
     print(y_predict)  # 打印结果
```

2.3 参数说明

支持生物序列机的模型包含很多参数，这些参数影响了模型的最终效果，我们提供了相关的用户接口以修改这些参数，具体说明如表 2-2 所示，只需要在创建 sbsm 对象时指定参数即可。

表 2-2 支持生物序列机的参数

参数名	说明	默认
c	支持向量机的罚分系数	64
alpha	用于核参数化的控制参数	-1
match_score	序列比对中的匹配得分	1
Mismatch_score	序列比对中的不匹配的罚分	-1
gap_score	序列比对中的间隙罚分	-2
k_neighbors	在 HCKDM 中，代表用于局部核心选择的邻近核心数	15
Lambda	在全局和局部核心对齐中，全局核心对齐的比率参数	0.9
nu1	拉普拉斯图正则化项的正则化参数	0.01
nu2	L2 正则化项的正则化参数	0.01

第 3 章　Linux 命令行工具

3.1 命令行工具的下载和安装

支持生物序列机提供了 Linux 系统的命令行工具，能够支持用户简洁快速地使用该软件，该工具仅支持 Linux 系统。

Linux 系统的命令行工具已经发布到 Conda 工具的 malab 的 channel 中，安装支持生物序列机需要先安装 Conda。Conda 的使用教程请参照附录 1。

使用 Conda 安装支持生物序列的命令如下所示：

```
1. conda install -c malab -c conda-forge sbsm
```

安装完成后，输入命令 sbsm，即可进入支持生物序列的主界面，主界面的示例如图 3-1 所示。

```
sbsm: Support Bio-sequence Machine

Version: 2.0        Contact: wyz020@126.com

http://lab.malab.cn/soft/SBSM-Pro/

Usage:
  Train:
    sbsm -t [options] <train_samples.fasta> <train_labels.txt> > results.model
  Predict:
    sbsm -p [options] <results.model> <test_samples.fasta> > predict_labels.txt

Options:
  Support vector machine with kernel parametered:
    -c  --c          FLOAT   Penalty coefficients for support vector machines. [64]
    -a  --alpha      FLOAT   Control parameter for kernel parameterization. [-1]
  Sequence alignment with Smith-Waterman algorithm:
    -m  --match      INT     match score [1]
    -x  --mismatch   INT     mismatch penalty [-1]
    -g  --gap        INT     gap penalty [-2]
  Mutiple kernel learning:
    -k  --kneighbors INT     In HCKDM, k represents the number of neighboring kernels chosen for local kernel selection. [15]
    -l  --lambda     FLOAT   Ratio parameter of the global kernel alignment in both global and local kernel alignments. [0.9]
    -n1 --nu1        FLOAT   Regularization parameter for the Laplacian graph regularization term. [0.01]
    -n2 --nu2        FLOAT   Regularization parameter for the L2 regularization term. [0.01]
  Others:
    -h  --help               Print this help usage information.
    -v  --version            Show version number.

(sbsmtolstoy) [redacted]:~$
```

图 3-1 支持生物序列的主界面

支持生物序列机的主页包含了版本信息，网址和参数说明。支持生物序列主要包含 Train 和 Predict 两个模式。

3.2 标准的输入文件

为了能够很好地描述数据内容，即蛋白质包含的氨基酸序列，同时，也要保证数据集的通用性和统一性，为此，应该将搜集到的数据以相关研究领域广为人知、应用广泛的格式或形式存取。如同公民身份证号，在国际研究中，每个蛋白质都包含一个唯一标识符，用以唯一表示该蛋白质，该唯一识别符也应该在数据文件中被体现。

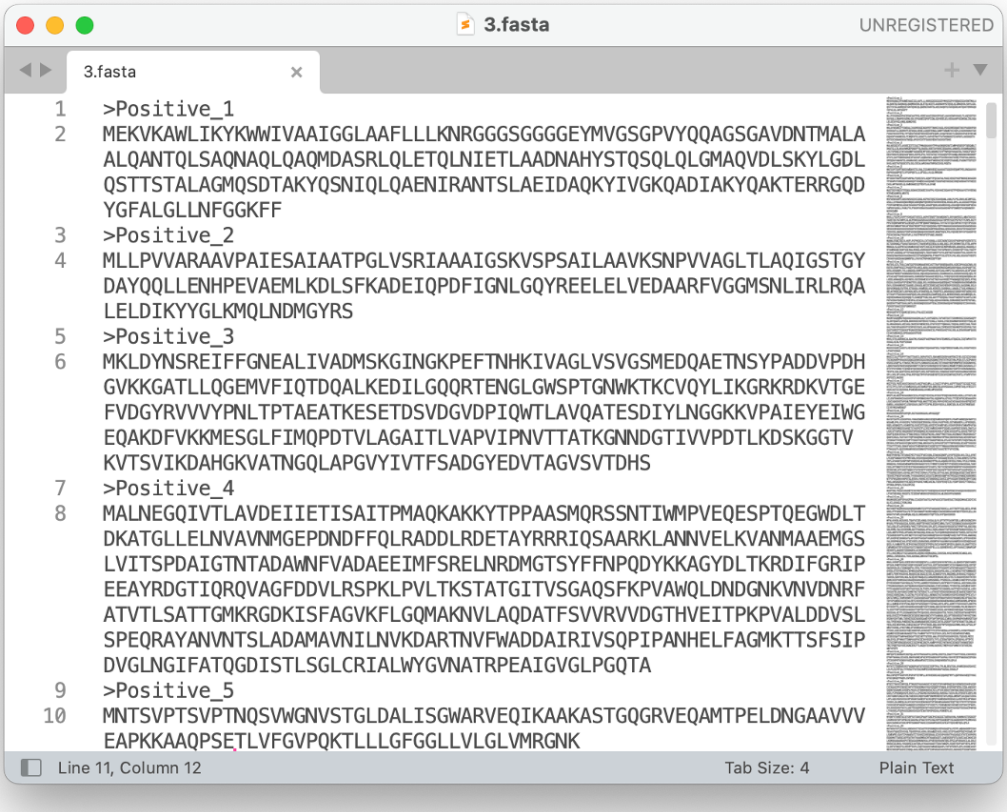


图 3-2 Fasta 格式文件示例

Fasta 格式是生物信息学中经常使用的一种表示核酸序列或蛋白质序列的文本格式。Fasat 格式就满足了完整描述蛋白质包含的氨基酸序列和标识蛋白质唯一标识符的条件，并且，通过以换行符隔开，多条蛋白质序列也可以在同一 Fasta 文件中保存。

以一条标准的蛋白质序列 Fasta 文件为例，该文件示例如图 3-2 所示，它包含蛋

白质标识符和具体序列两部分，并通过换行符隔开。在标准的蛋白质序列 Fasta 文件中，第一行的第一个字符均是符号“>”，该符号后面为序列标识符和描述信息，这些信息均用于标记序列，且每个序列的标识都具有唯一性，从第二行开始为具体序列内容。序列行的字母代表了组成蛋白质的氨基酸，它们之间的对应关系如表 3-1 所示。

表 2-1 蛋白质序列中的 20 种氨基酸

氨基酸名称	字母缩写表示	氨基酸名称	字母缩写表示
丙氨酸	A	甲硫氨酸	M
半胱氨酸	C	天冬氨酸	N
天冬氨酸	D	脯氨酸	P
谷氨酸	E	谷氨酰胺	Q
苯丙氨酸	F	精氨酸	R
甘氨酸	G	丝氨酸	S
组氨酸	H	苏氨酸	T
异亮氨酸	I	缬氨酸	V
赖氨酸	K	色氨酸	W
亮氨酸	L	酪氨酸	Y

3.3 使用流程

支持生物序列机的使用主要包含两个步骤，具体细节如下：

（1）使用训练集训练模型。训练集包含两个部分，分别是以 Fasta 格式存储的蛋白质序列和以 txt 格式存储的对应的标签。输入蛋白质序列和对应的标签可以训练一个支持生物序列机的模型，这一步使用 sbsm 的 Train 模式，具体命令如下：

```
1. sbsm -t train_samples.fasta train_labels.txt
```

其中 train_samples.fasta 是你的蛋白质序列的样本文件，每两行表示一个序列；train_labels.txt 是你的标签文件，每一行对应一个序列标签。

sbsm 的训练模式会生成一个 results.model 文件，这个文件在后续预测中使用。

（2）使用测试集测试模型。测试集仅包含被测试的蛋白质序列，它也是以 Fasta

格式存储的。此外，还需要指定测试使用的模型，即上一步得到的 results.model。这一步使用 sbsm 的 Predict 模式，具体命令如下：

```
1. sbsm -p results.model test_samples.fasta
```

其中 results.model 是你的在上一步训练得到的模型文件；test_samples.fasta 是你的蛋白质序列的样本文件，每两行表示一个序列。sbsm 的预测模式会生成一个 txt 文件，该文件包含了对测试样本的标签的预测结果，与真实标签进行对比，可以评估模型的效果。

3.3 模型参数

支持生物序列机的模型包含很多参数，这些参数影响了模型的最终效果，我们提供了相关的用户接口以修改这些参数，具体说明如表 2-2 所示：

表 2-2 支持生物序列机的参数

短命令	长命令	说明	默认
-c	--c	支持向量机的罚分系数	64
-a	--alhpa	用于核参数化的控制参数	-1
-m	--match	序列比对中的匹配得分	1
-x	--mismatch	序列比对中的不匹配的罚分	-1
-g	--gap	序列比对中的间隙罚分	-2
-k	--kneighbors	在 HCKDM 中，代表用于局部核心选择的邻近核心数	15
-l	--lambda	在全局和局部核心对齐中，全局核心对齐的比率参数	0.9
-n1	--nu1	拉普拉斯图正则化项的正则化参数	0.01
-m2	--nu2	L2 正则化项的正则化参数	0.01
-h	--help	打印帮助使用信息	NaN
-v	--version	显示 sbsm 的版本号	NaN

如果用户希望指定这个参数，只要在 Train 模式中指定短命令或者长命令，然后加一个空格，并指定相关数值即可。例如，在训练模型时希望指定支持向量机的罚分系数为 32，序列比对中的匹配得分 2，则使用的命令（其中标红的部分即是指定参

数）如下所示：

```
1.  sbsm -t train_samples.fasta train_labels.txt -c 32 -m 2
```

附录 1 Conda 教程

1.1 Conda 概述

在服务器上使用 Linux 命令行安装 Conda（Conda 可以理解类似于应用商店或是 mac 里的 App Store。可以在 conda 里面安装软件，或者在 conda 之外安装），使用 conda 管理小环境和使用 conda 管理软件，用 conda 来安装和管理生信软件以及环境比较方便。

Conda 包含 miniconda 和 anaconda。miniconda 比较简单，只能在命令行中使用，anaconda 比较强大，有一个界面化的软件，但是占用系统空间比较大。现在使用 Linux，用命令行操作的 miniconda 就可以了。Conda、miniconda 和 anaconda 的之间的关系如图 1 所示。

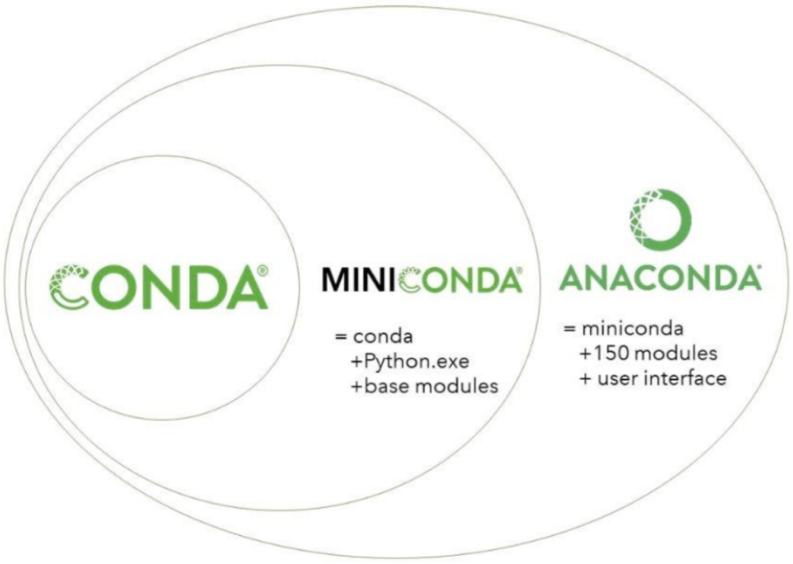


图 1 Conda、miniconda 和 anaconda 的关系图

1.2 Conda 下载和安装

使用 Linux 系统中的 wget 工具，可以下载 miniconda，命令如下所示：

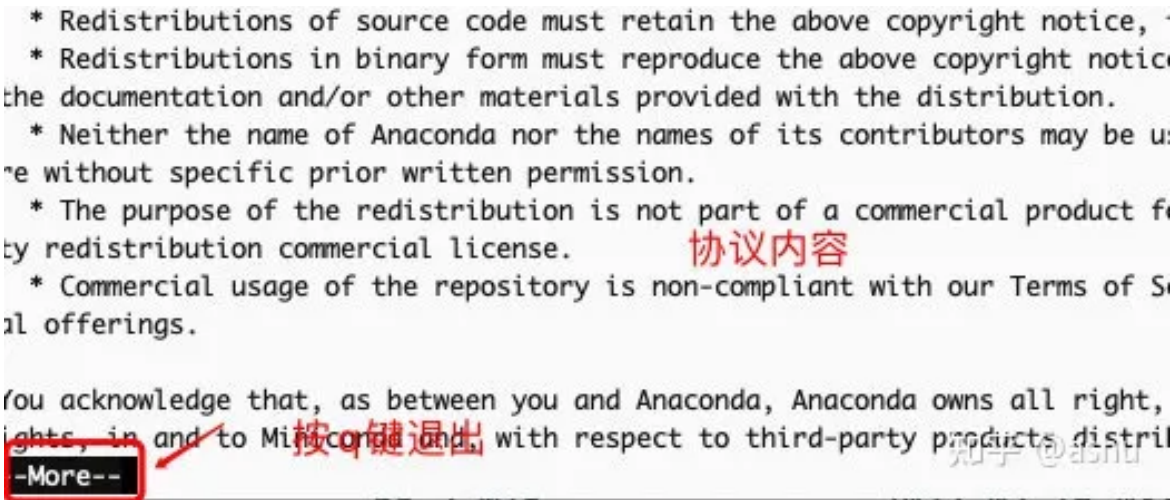
```
1.  wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

下载完成后，使用 bash 安装 miniconda，命令如下所示：

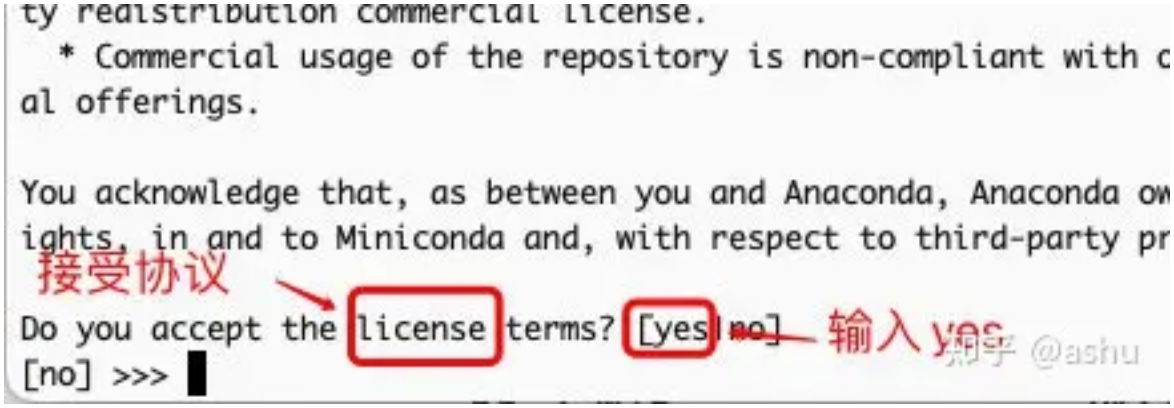
```
1.  bash Miniconda3-latest-Linux-x86_64.sh
```

安装过程有多个提示信息，需要按 Enter (回车键)或者输入 yes，（如果输入 yes 时，不小心输多了，就按 control 和退格键删除），安装过程主要如下所示：

（1）看到 more 就是按空格键翻页查看协议，按 q 退出



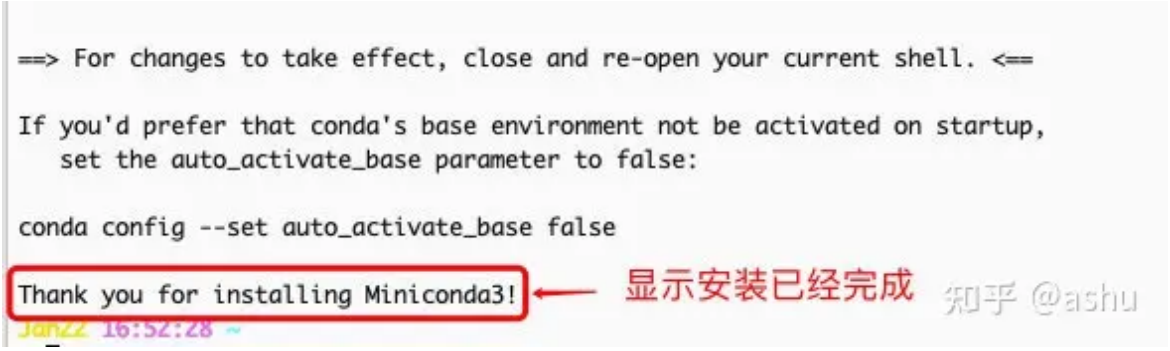
（2）接受协议，输入 yes



（3）默认安装路径，按 enter

（4）会询问是否需要初始化，输入 yes

（5）显示安装已完成的提示信息



（6）激活 Conda 资源，命令如下所示：

```
1.  source ~/.bashrc
```

1.3 Conda 换源

使用 conda 是需要它去安装其它软件（如生信软件），conda 是默认去自己的官网搜索，而我们使用的服务器是在国内，conda 的网在国外，从国内的网络去访问国外的网络就是特别的慢，所以需要配置镜像，如配置清华的镜像。

下面这四行配置清华大学的 conda 的 channel 地址，国内用户推荐，使用命令如下所示：

```
1. conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
2. conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
3. conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
4. conda config --set show_channel_urls yes
```

1.3 Conda 的使用

Conda 的命令中主要包含以下几个大类：

（1）创建新环境

conda create --name env_name

conda create --name env_name python=3.5 # 创建指定 python 版本

conda create --name env_name python=3.5 numpy scipy # 创建指定 python 版本下包含某些包

（2）激活/使用/进入某个虚拟环境

activate env_name

（3）退出当前环境

deactivate

（4）复制某个虚拟环境

conda create --name new_env_name --clone old_env_name

（5）删除某个环境

conda remove --name env_name --all

（6）查看当前所有环境

conda info --envs 或者 conda env list

（7）查看当前虚拟环境下的所有安装包

conda list 需进入该虚拟环境

conda list -n env_name

（8）安装或卸载包(进入虚拟环境之后）

conda install xxx

conda install xxx=版本号 # 指定版本号

conda install xxx -i 源名称或链接 # 指定下载源

conda uninstall xxx

（9）分享虚拟环境

conda env export > environment.yml # 导出当前虚拟环境

conda env create -f environment.yml # 创建保存的虚拟环境

（10）批量导出虚拟环境中的所有组件

conda list -e > requirements.txt # 导出

conda install --yes --file requirements.txt # 安装