

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

import os                #导入标准库 os
import re
import sys
import gzip
import logging
import argparse

LOG = logging.getLogger(__name__)

__version__ = "1.1.0"      #设置版本信息
__author__ = ("Xingguo Zhang",)    #输入作者信息
__email__ = "invicoun@foxmail.com"
__all__ = []

#python 缩进为 4 个空格，赋值后面可以空一行，函数之前空两行
def read_fastq(file):      #构建函数（子函数）
    """Read fastq file"""  #对函数进行说明

    LOG.info("Reading message from %r" % file)    #显示日志
    if file.endswith("fastq.gz") or file.endswith(".fq.gz"):    #通过文件尾缀判断文件格式
        fp = gzip.open(file)    #使用 gzip 打开压缩文件
    elif file.endswith(".fastq") or file.endswith(".fq"):
        fp = open(file)    #打开正常文件
    else:
        raise Exception("%r file format error" % file)    #输入文件的格式错误时程序报错

    seq = []
    for line in fp:
        if isinstance(line, bytes):    #判断数据的格式
            line = line.decode('utf-8')
        line = line.strip()

        if not line or line.startswith("#"):
            continue
        if not seq:
            seq.append(line.strip("@").split()[0])
            continue

        seq.append(line)
        if len(seq) == 4:
            yield seq    #使用 yield 函数生成迭代器，节约存入字典或者列表消耗的内存
            seq = []

    fp.close()

def fq2fa(files, minlen):    #运行函数

```

```

'''Convert fastq files to fasta files'''

for file in files:
    for seqid, seq, ignore, quality in read_fastq(file):
        if len(seq) < minlen:
            LOG.info("The length of the filter sequence %s is %s" % (seqid, len(seq)))
            continue
        print('>%s\n%s' % (seqid, seq))

return 0

def add_hlep_args(parser):    #帮助函数

    parser.add_argument('fastq', nargs='+', metavar='FILE', type=str,
        help='Input fastq file.')
    parser.add_argument('--minlen', metavar='INT', type=int, default=0,
        help='Set the minimum length of sequence filtering, default=0')
    return parser

def main():    #主函数，执行函数

    logging.basicConfig(
        stream=sys.stderr,
        level=logging.INFO,
        format="[% (levelname)s] %(message)s"
    )

    parser = argparse.ArgumentParser(formatter_class=argparse.RawDescriptionHelpFormatter,
        description=""
name:
    fq2fa.py -- Convert fastq files to fasta files    #说明脚本的功能

attention:    #列举脚本的用法
    fq2fa.py data.fastq > data.fasta
    fq2fa.py data.fastq --minlen 500 > data.fasta
version: %s
contact:  %s <%s>\    #在帮助的时候显示版本和作者信息
    ''' % (__version__, ' '.join(__author__), __email__))

    args = add_hlep_args(parser).parse_args()

    fq2fa(args.fastq, args.minlen)

if __name__ == "__main__": #固定格式，使 import 到其他的 python 脚本中被调用（模块重用）执行
    main()

```