structural\_protein\_package.py脚本使用说明文档。

函数名称、参数列表及作用介绍：

1. getGI(s)

参数：字符串s

返回值：一个字符串，代表从s中提取到的GI信息

作用：提取GI信息  
2. ConvertXlsToGIFIle(XlsFilename)

参数：字符串XlsFilename，代表存有数据信息的excel表格文件的地址，默认为本目录下的Dataset\_S5.xls文件

返回值：无

作用：从XlsFilename文件中提取两类蛋白质的GI数据并分别写入本目录下的posSetGI.txt和negSetGI.txt文件中

3. getEntry(gi)

参数：字符串gi，代表某个蛋白质的GI信息

返回值：字符串que，代表所查询的蛋白质的序列信息

作用：根据传入的GI信息查询相应的蛋白质序列并返回。**注意：该函数只能在安装有blast数据库的Linux系统中调用，否则无效。**

4. convertGItoSeq(posGIFilename,negGIFilename)

参数：posGIFilename和negGIFilename，分别代表存储有两类蛋白质的GI信息的文件地址，默认为本目录下的posSetGI.txt文件和negSetGI.txt文件。

返回值：无

作用：从posGIFilename和negGIFilename中读取两类蛋白质的GI信息，并利用getEntry函数查询到这些蛋白质的序列并存储至本目录下的posSet.txt和negSet.txt文件中。**注意：该函数只能在安装有blast数据库的Linux系统中调用，否则无效。**

5. getVec(s)

参数：字符串s。s为一个蛋白质序列（大写字母组成的字符串）

返回值：一个含有20个元素的list。表示蛋白质序列s中的20种氨基酸的频率，即返回一个20维的向量，向量每一维度代表的氨基酸与Amino\_Table中的氨基酸顺序相同。

作用：计算蛋白质序列s中的氨基酸频率

6. data\_processing(posSetFilename,negSetFilename)

参数：posSetFilename和negSetFilename是两个文件名，分别代表了存有两种蛋白质序列的文件。默认为本目录下的posSet.txt文件和negSet.txt文件

返回值：四个list，分别是Vec,Y,posTestset,negTestset。分别代表了：Vec：所有蛋白质的氨基酸频率向量的集合；Y：每个蛋白质的标签，0或1，顺序与Vec中蛋白质的顺序相对应；posTestset：阳性蛋白质的测试集；negTestset：阴性蛋白质的测试集。

作用：根据posSet.txt文件和negSet.txt文件中存储的蛋白质序列的信息，计算所有蛋白质的氨基酸频率并且根据不同的作用、分类存储进Vec,Y,posTestset,negTestset中。

7. classify(Vec,Y)

参数：两个list，Vec和Y。Vec是蛋白质氨基酸频率向量的集合，Y是Vec中每个蛋白质对应的标签。

返回值：clf，一个sklearn分类器。该分类器是根据传入的Vec和Y建立、训练的，结构为ANN。

作用：根据给定数据构建ANN模型。

8. predict(clf,predictsetFilename)

参数：clf和predictsetFilename，clf是一个已经训练好的sklearn分类器，predictsetFilename是一个文件名，代表存储了需要预测的蛋白质序列的文件，默认为本目录下的predictset.txt文件。

返回值：无

作用：利用分类器clf对predictsetFilename中每个蛋白质种类进行预测，预测的结果存储在本目录下的predictresult.txt文件中。

9.testing(posPre,negPre)

参数：两个list，posPre和negPre。posPre存储了阳性蛋白质氨基酸频率向量的预测结果，negPre存储了阴性蛋白质氨基酸频率向量的预测结果。

返回值：xx,yy,ACC,MCC。xx是一个list，存储了FPR值。yy是一个list，存储了TPR值。其中xx列表和yy列表中的数据都是一一对应的。ACC是一个变量存储了分类的准确率，MCC是一个变量，存储了分类的马修斯相关系数。

10. get\_k\_fold\_Cross\_Validation\_classifier(Vec,Y,k)

参数：Vec,Y,k。Vec是一个list，存储了所有蛋白质氨基酸频率向量。Y是一个list，存储了Vec中每一条蛋白质数据对应的标签。k代表k折交叉验证中的参数k。

返回值：clfs和testset。clfs是一个list，里面存储了k个已经训练好的sklearn分类器，这是k折交叉验证过程中建立的k个分类器。testset是一个list，里面存储了这k个分类器每一个分类器所用的测试集（每个测试集包含了测试用的蛋白质氨基酸频率向量以及蛋白质对应的标签）。

作用：实现k折交叉验证。返回k折交叉验证中用到的k个分类器和这些分类器所用的测试数据。

11. calcACC(clf,testx,testy)

参数：clf,testx,testy。clf是一个已经训练好的sklearn分类器。testx和testy是两个list，分别存储了测试该分类器用的蛋白质氨基酸频率向量以及蛋白质对应的标签

返回值：一个数字：ACC。代表分类器clf在该测试集上的准确率。

作用：计算指定分类器在指定测试集上的运行准确率。

12. calcFPR\_TPR(clf,testx,testy)

参数：与calcACC函数相同

返回值：xx,yy,ACC,MCC,TP,FP,TN,PN。xx和yy与前面testing函数返回值中的xx与yy的作用是相同的，存储了对应的FPR与TPR的值。ACC为准确率，MCC为马修斯相关系数，TP、FP、TN、PN分别代表了分类中的真阳性、假阳性、真阴性、假阴性的数量。

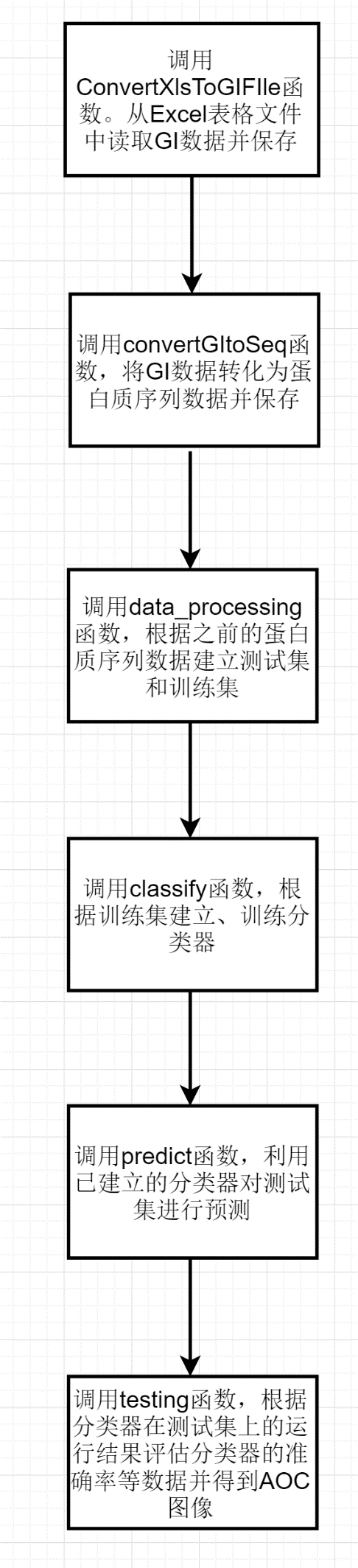
13. k\_fold\_Cross\_Validation\_classifier(X,Y,k)

参数：X,Y,k。X是一个list，存储了所有蛋白质氨基酸频率向量。Y是一个list，存储了X中每一条蛋白质数据对应的标签。k代表k折交叉验证中的参数k。

返回值：一个数字ACC，代表了在该数据集上运行k折交叉验证后得到的模型准确率。

作用：计算在数据集（X,Y）上运行k折交叉验证后得到的模型准确率。

**算法流程图：**



**脚本使用教程：直接import task1\_package.py，然后就可以调用task1\_package.py中的函数了。**