目录

一．实习内容分析

1.1 引言

生物物理学（Biological Physics）是物理学与生物学相结合的一门交叉学科，是生命科学的重要分支学科和领域之一。生物物理学是应用物理学的概念和方法研究生物各层次结构与功能的关系、生命活动的物理、物理化学过程和物质在生命活动过程中表现的物理特性的生物学分支学科。生物物理学旨在阐明生物在一定的空间、时间内有关物质、能量与信息的运动规律。

生物物理学的不断发展和完善，一定会极大地促进生命科学的发展，并将带来对于生命现象的本质新的突破。二十一世纪是生命科学的世纪，更是学科交叉、科学走向统一的世纪。新的世纪留给生物物理学的任务有：

（1）发掘非平衡开放系统特性的主要规律，也就是找出生命的热力学基础。

（2）从理论上解释进化和个体发育的现象。

（3）解释自身调节和自我复制的现象（自组织现象）。

（4）从原子、分子水平上揭露生物过程的本质也就是找到活跃在细胞内的蛋白质、核酸及其他物质的结构和生物功能的联系；此外，还要在研究生命体在更高的超分子水平上、在细胞的水平上及在构成细胞的细胞器的水平上的物理现象。当然，这些都需要化学的帮助与支持。

（5）设计出研究生物功能物质及由这类物质构成的超分子结构的物理方法和物理化学方法，并对利用这种方法所得到的结果提供理论解释。

（6）对神经脉冲的发生和传播、肌肉收缩、感觉器官对外部信号的接收及光合作用等高度复杂的生理现象，提供物理的解释。

（7）解释怎样由物质形成了意识。

1.2 实习目的

（1）暑期实习是本科生教育重要的实践性教学环节，是学生在学习基本理论基础之后的一个综合性实践环节。强调培养实践学生的应用能力、达到理论联系实际的目的。通过实习提高学生的综合能力；提高学生独立思考、分析问题和独立工作的能力。通过实习，巩固学生所学的专业理论知识，加深对所学基础知识及专业理论的理解，进一步巩固和扩大专业知识面，锻炼学生在该学科领域发现问题、分析问题、解决问题及实际动手能力，培养学生劳动意识，为后续课程的学习和走向打下坚实的基础。

（2）通过暑期实习，根据所学到的知识、掌握的研究方法、对该领域前沿发展的了解等，为自己的毕业设计做好准备，了解毕业设计中所涉及到的主要功能、主要内容以及主要的模块划分，通过阅读相关书籍文献的学习，为毕业设计顺利进行做好充分的准备。

1.3 实习地点

细胞是由大量的生物大分子（蛋白质、核酸和糖分子）以及小分子相互作用形成的分子网络。研究这些分子以及它们形成的网络的结构与动力学，是理解细胞活动规律的基础。生物物理研究组致力于用计算机模拟方法研究生物分子的结构、动力学、相互作用以及网络，建立理论模型，揭示基本规律。而我在刘士勇老师的指导下，在科技楼 5 楼生物物理组研究生实验室进行为期两周的实习。

1.4 实习内容

（1）了解蛋白质相互作用预测相关方面的理论知识，看懂《基于序列的蛋白蛋白相互作用预测》论文，并按照论文中给出的理论方法及步骤，及其提供的蛋白质序列数据，自己完成代码，通过运行程序得出结果并分析。

（2）学会如何获取蛋白质 DNA 等数据，以及生物方面各种数据库的使用。

（3）了解生物物理领域各大期刊文献，学会如何查找文献，阅读与其相关的英文文献，熟悉相关领域的英文单词，了解该领域目前的发展情况。

（4）安装生物物理研究方面所需环境和工具，熟悉各种软件的作用和功能。

二．专题内容分析

2.1 蛋白质相互作用预测

2.1.1 研究意义

蛋白质是生命的物质基础，也是人体六大营养元素之一，人体的每一个细胞和所有重要组成部分都有蛋白质的参与，许多生理过程都和蛋白质分子之间或蛋白质分子与其它分子之间的相互作用相关，例如催化反应，免疫反应，肌肉运动，信号传递等。因此，对蛋白质的深刻研究是必要的。

蛋白质-蛋白质相互作用（PPI）是指蛋白质之间的相互关联关系，如：两个蛋白质分子之间是否发生接触，某些细胞过程等。在细胞活动中，如蛋白质折叠、蛋白质转录以及翻译和翻译后修饰等，蛋白质-蛋白质相互作用都扮演着重要角色，并且绝大多数蛋白质是通过和其它蛋白质之间的相互作用来完成蛋白质功能。所以，研究蛋白质相互作用对增进了解蛋白质功能有着重要意义。

生物实验方法为早期的主要方法，该方法主要分为小规模实验和高通量实验。小规模实验方法目的性强并且正确率高，但由于每次测量的相互作用数目少，导致很难使用此方法构建蛋白质相互作用网络。现有的小规模实验方法主要有：蛋白质亲和色谱、免疫共沉淀、荧光共振能量转移、X 射线晶体学、亲和印证、核磁共振、标签转移等。随着科技发展，高通量的大规模实验方法纷纷出现，主要有：串联亲和纯化、酵母双杂交实验、蛋白质芯片。高通量实验方法相比较于小规模实验方法，优点在于更加简 便高效。但是，虽然高通量实验方法比小规模方法产生数据量大，速度快，但是由于各种误差因素导致高通量的方法产生大量假阴性和假阳性数据，并且不同的实验结果重叠率低。因此，高通量的方法必须经过小规模方法进行验证，才能保证结果的准确性。但是，实验方法不仅只能覆盖蛋白蛋白相互作用网络的一小部分，假阳性率高，而且实验周期和试验费用

都较高。所以，发展新的有效地自动计算方法对蛋白质相互作用网络的建立是十分必要的。

2.1.2 研究现状

（1）Bock与Gough在 2001年提出了基于序列的方法预测蛋白质相互作用。这是出现较早的方法。他们通过考虑蛋白质序列中的氨基酸残基的理化属性电荷、残基表面张力和疏水性，对蛋白质序列进行编码然后得到特征向量，之后用SVM 预测蛋白质的相互作用。

（2）Shen等提出了编码三联体组合信息的方法来表示蛋白质序列。首先根据氨基酸侧链的体积和偶极特性将氨基酸分成7组，然后把相邻的三个残基作为整体，之后通过统计序列中三联体的分布情况来表示蛋白质序列。之后用结合S-核函数的 SVM 对人类蛋白质的相互作用进行了预测，其正确率达到 83.9%。

（3）Guo等在充分考虑蛋白质序列内部的氨基酸之间长程相互作用，提出了使用自协方差的编码式表示来蛋白质序列，使用基于径向基-核函数的 SVM 方法对酵母蛋白质的相互作用进行了预测，其正确率达到 88.09%。

（4）Xia 把 Moran 的自相关描述符和旋转森林分类器两种算法结合起来预测蛋白质的相互作用。自相关描述符是考虑蛋白质序列中近邻信息，它可以刻画蛋白质序列的内部短程与长程相互作用，所以能够很好的揭示整个蛋白质序列上和蛋白质的相互作用有关模式。旋转森林，是一种集成的分类器，把它与自相关描述符相结合，能够有效地提高蛋白质的相互作用预测水平。

蛋白质之间的相互作用经常发生于序列上的间断片段之间，但是相距较远的残基在折叠的空间上有可能也会相距很近。在考虑了相互作用的间断性，以及结合 Shen 的方法，Yang 等提出了使用分段局部的蛋白质序列表示描述符预测蛋白质的相互作用。史明光使用了相关系数来编码蛋白质序列，同时又考虑了序列内部的长程相互作用以及序列之间协同进化的关系。Ding 等提出了使用氨基酸索引分布的蛋白质序列表示法。

集成学习的方法也被使用于蛋白质的相互作用预测。夏俊峰等人将通过六种不同的蛋白质序列表示方法得到的 SVM 分类器进行了集成，使用已经得到的集成分类器在酵母的蛋白数据集上进行了蛋白质相互作用的预测，其正确率为90.81 %。

由于蛋白质特征向量的维度不同而且较高。所以不同的降维方法和对数据不平衡性的处理在预测相互作用中也有体现。上面所提到的的序列表示方法编码后所得到的特征向量通常维数比较高，Zhang等人使用原始信号的稀疏性，再利用压缩感知算法用于特征向量的降维，提出了新的预测蛋白质相互作用的方法-适应压缩学习，在这之后，他们与多种特征选择和降维算法，例如因子分析，局部保持投影， Fisher 打分，随机接近嵌入等进行了比较，最后对蛋白质相互作用的不同方法的预测情况进行了讨论。

2.1.3 研究内容

典型的方法用于构造基于序列的 PPI 预测模型有两个主要步骤：（1）提取离散向量表示蛋白质序列的功能；（2）培养一个高效的在特征空间中的分类算

法。

主要使用的序列离散模型包括：特定位置打分矩阵；伪氨基酸组；dipeptide composition；二肽组成；并联合分析功能。

不同的统计和打分分类方法应用于 PPI 的预测中，如贝叶斯模型，支持向量机（SVM），Logistic 回归，以及集成方法。

2.1.4 研究方法

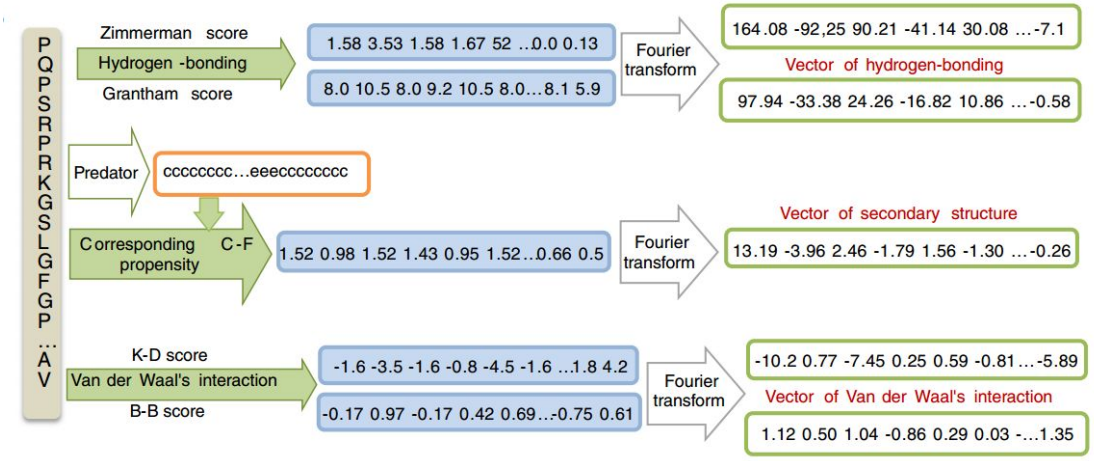
现有的基于序列的预测模型可以大致分为两类：（1）结构域或基序为基础的方法及（2）连续离散模型为基础的方法。 对于模型第一组，蛋白质是通常由几个结构域代表。 Huang 等人提出了一个名为基于最大域的特异性集体封装方法（MSSC）预测蛋白蛋白相互作用，他的算法应用于就算复杂的蛋白质结构域集合问题，它代表蛋白质的相互作用的状态和它们对应的间域结构之间的关系。通过利用蛋白质相互作用数据和域信息，MSSC 取得了相对较好的结果。 Pitre 等人还开发了称为 PIPE 的基于模体的方法。在确定蛋白质对是否有相互作用，PIPE 搜索出现在蛋白质对那些已经公知的 PIPE 子序列。有两个主要问题存在于 PIPE：高计算费用和相对较低的特异性。 虽然 PIPE 已经提出的改进版本，它的效率在许多基序存在于一级序列时仍是有限的。Dong 等人提出了基于蛋白质的四个基本构建序列的方法。综合四种不同的模块被证明优于任何独立的模块，它可以被视为一种基于域的方法和模块合并的组合方法。对于第二组，蛋白质的氨基酸序列通常被编码为一个离散的矢量，然后进行预测的算法。Shen 等人认为三肽中三条连续的氨基酸作为一个单元表示离散模型，然后蛋白质对是 686 维量。而这种离散模型用于展示相邻氨基酸情况，氨基酸之间具有较长的距离的情况被忽略。Rao 等人提出了氨基酸残基的关联模型来预测蛋白蛋白相互作用，主要有两个步骤：特征编码，建立线性关系模型。其他预测蛋白蛋白相互作用的方法按时间顺序还包括：伪氨基酸组成的 discete 模型和自相关化描述符。

2.2 数据集构建

选取人类蛋白的阴性数据以及酵母细胞的阳性数据作为研究对象，获取蛋白质 ID 之后在 UNIPROT 中进行序列搜索，由于一些蛋白质被删除，我们把这些蛋白质从我们的数据集中删除，之后使用 CD—HIT 程序对数据集去冗余，选择参数 30%。剩余的蛋白质对数为 2716 对。

2.3 特征值选取

编码序列为特征向量，我们使用了范德华相互作用，氢键特性的各两种方法，以及使用二级结构的方法对蛋白质序列进行编码，每个氨基酸残基对应一个特征值。如下图：



Predator 软件被用来预测蛋白质的二级结构，并使用 Chou-Fasman 的方法对每个氨基酸残基进行编码。通过将每个氨基酸代替为相应的序列中 Fasman 值，蛋白质序列转化成一个数字特征矢量。

氢键的特征向量使用 Grantham 的方法和 Zimmerman 的方法被编码。范德华特征向量采用 Kyte-Doolittle 的方法和 Bull-Breese 的方法进行编码。所以，连同蛋白质二级结构的特征向量，每个蛋白序列被编码成 5 个数值特征向量。

氨基酸	A	L	R	K	N	M	D	F	C	P
范德华	1.8	3.8	-4.5	-3.9	-3.5	1.9	-3.5	2.8	3.5	-1.6
氢键 1	0	0.13	52	49.5	3.38	1.43	49.7	0.35	1.4	1.58
氢键 2	8.1	4.9	10.5	11.3	11.6	5.7	13	5.2	5.5	8
二级结构	0.66	0.59	0.95	1.01	1.56	0.6	1.46	0.6	1.19	1.52
极性	8.1	4.9	10.5	11.3	11.6	5.7	13	5.2	5.5	8

氨基酸	Q	S	E	T	G	W	H	Y	I	V
范德华	-3.5	-0.8	-3.5	-0.7	-0.4	-0.9	-3.2	-1.3	4.5	4.2
氢键 1	3.53	1.67	49.9	1.66	0	2.1	51.6	1.61	0.13	0.13
氢键 2	10.5	9.3	12.3	8.6	9	5.4	10.4	6.2	5.2	5.9
二级结构	0.98	1.43	0.74	0.96	1.56	0.96	0.95	1.14	0.47	0.5
极性	10.5	9.2	12.3	8.6	9	5.4	10.4	6.2	5.2	5.9

2.4 维度转换

每个蛋白序列序列分别转化为五个数字特征向量。 然而，这些特征向量不能用于直接计算，因为每个维度矢量依赖于相应的蛋白质序列的长度，这使得它不可能找到一个固定矩阵 M 来进行计算。 因此，为了统一，向量需要进行维度的转换。

我们使用傅立叶级数来解决这个问题， 该公式列示如下：

$$X_k^{'} = \sqrt{\frac{2}{L}} \sum_{n=0}^{L} X_n \cos[\frac{\pi}{L}(n+\frac{1}{2})(k+\frac{1}{2})], k = 0,1,...,9$$

其中 L 是原始特征向量的长度。在这里，我们使用前 10 项的傅里叶级数作为新的数值特征向量。 长度选取为 10 的原因主要是当我们继续提高维度，计算结果并没有显著改善，而且随着维度的提高，计算速度会大大下降。

2.5 Fisher 判别法

维数问题是在应用统计方法中解决模式识别问题的主要困难之一，在低维空间有效的方法，在高维空间常常无效。所以，降低维数便成为了解决实际问题的关键因素。Fisher 判别法，就是涉及到维数压缩投影分类的方法。如果要把高维的模式样本从特征向量空间中投影到一条直线上，也就是把高维特征空间压缩成一维，这在数学的投影方法上很容易办到，但问题的关键是在投影之后，原来的线性可分样本有可能变得混杂在一起从而无法区分。但是，在一般情况下，总是可以找到一个最好的投影方向，使得样本投影在这个方向上，其一维数值是最容易分得开的。因此，如何找到这个最好的直投影方向，如何实现从高维到低维，其最投影好方向的变换，是 Fisher 判别法要解决的核心问题。我们以二维空间为例，我们将平面上的点投影到直线上，使两类样本投影后的值类间尽可能分散，类内尽可能聚集。

对于每个对特征矢量 P1 和 P2（P1 和 P2 分别代表一个蛋白质对中的两个蛋白质），我们希望训练一个矩阵 M，并使用<p | M |p'>的评分测量两个蛋白质之间的相互作用。

向量 X 是每一对蛋白质的特征向量相乘的结果，因为选取的傅里叶特征向量的长度为 10，因此 X 便是长度为 100 的向量，每一对蛋白质在选取某一个特定特征之后都会转化成一个 X。

为了将蛋白质阴性数据和蛋白质阳性数据分离开来，我们在训练之中将它们分别处理可以得到两类由 X 组成的样本向量。在此记为 X1 和 X2，X1 代表阳性数据，X2 代表阴性数据。

（1）由 $M_i = \frac{1}{n_i} \sum_{x_k \in X_i} x_k, i = 1,2$，计算 M_i。

（2）由 $\vec{S_i} = \sum_{x_k \in X_i} (x_k - M_i)(x_k - M_i)^T$ 计算各类的类内离散度矩阵 $\vec{S_i}, i = 1,2$

（3）计算类内总离散度矩阵 $\vec{S_w} = \vec{S_1} + \vec{S_2}$。

（4）计算 $\vec{S_w}$ 的逆矩阵 $\vec{S_w}^{-1}$。

（5）由 $w^* = \vec{S_w}^{-1}(M_1 - M_2)$ 求解 w^*。

2.6 ROC 曲线

ROC 曲线是受试者工作特征曲线的缩写。它以真阳性率（灵敏度）为纵坐标，假阳性率（特异度）为横坐标绘制的曲线。传统的评价方法有一个共同的特点，必须将试验结果分为两类，再进行统计分析。 ROC 曲线的评价方法与传统的评价方法不同，无须此限制，而是根据实际情况，允许有中间状态，可以把试验结果划分为多个有序分类，如正常、大致正常、可疑、大致异常和异常五个等级再进行统计分析。因此， ROC 曲线评价方法适用的范围更为广泛。

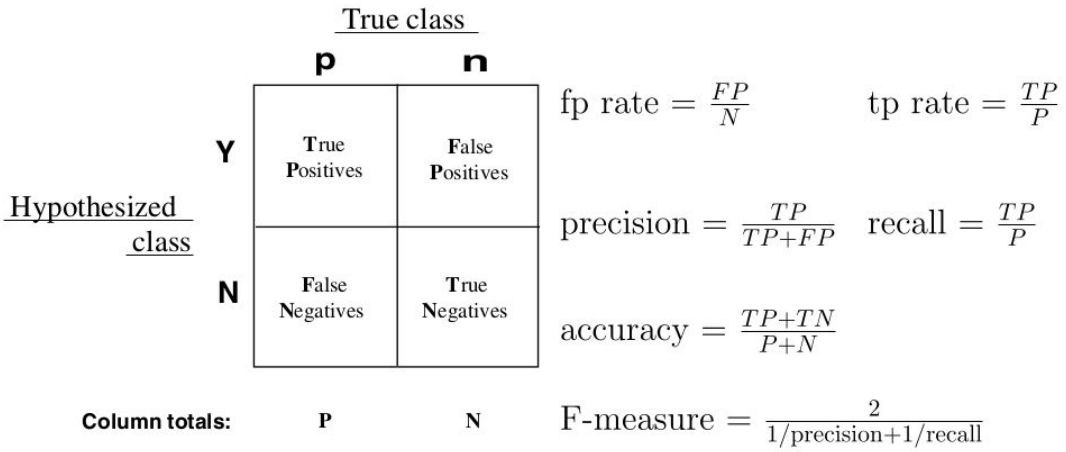


Fig. 1. Confusion matrix and common performance metrics calculated from it.

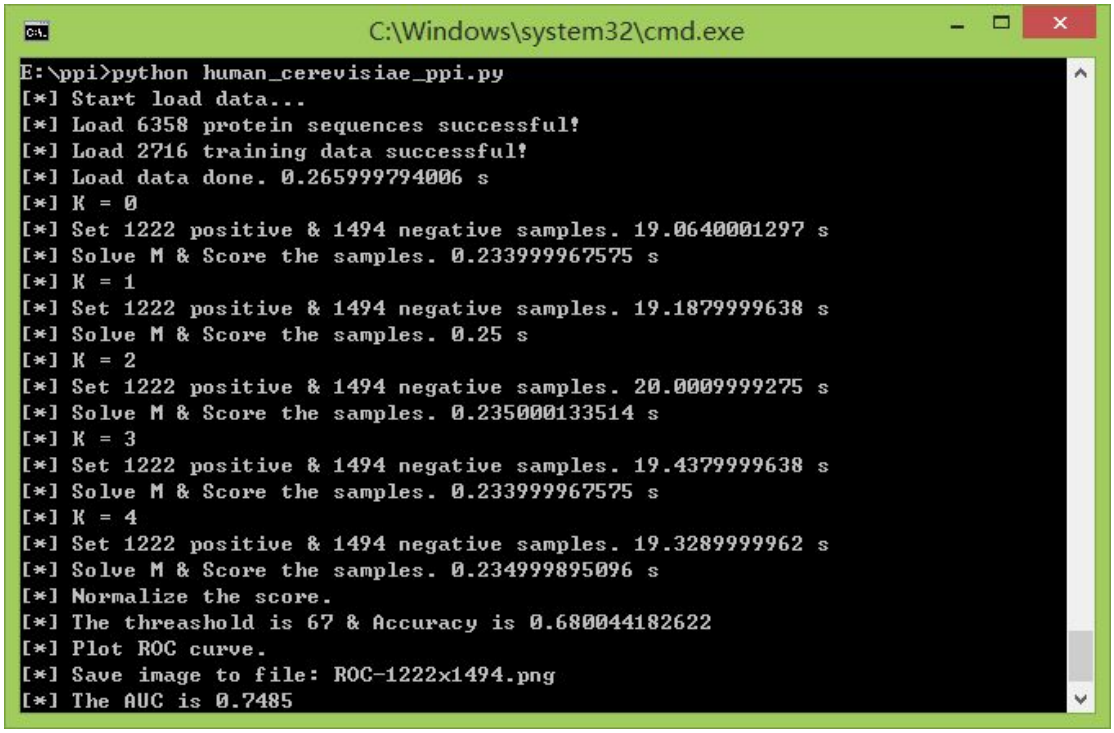
ROC 曲线是反映敏感性和特异性连续变量的综合指标,是用构图法揭示敏感性和特异性的相互关系，它通过将连续变量设定出多个不同的临界值，从而计算出一系列敏感性和特异性，再以敏感性为纵坐标、特异性为横坐标绘制成曲线，曲线下面积越大，诊断准确性越高。在 ROC 曲线上，最靠近坐标图左上方的点为敏感性和特异性均较高的临界值。

2.7 实验结果及程序代码

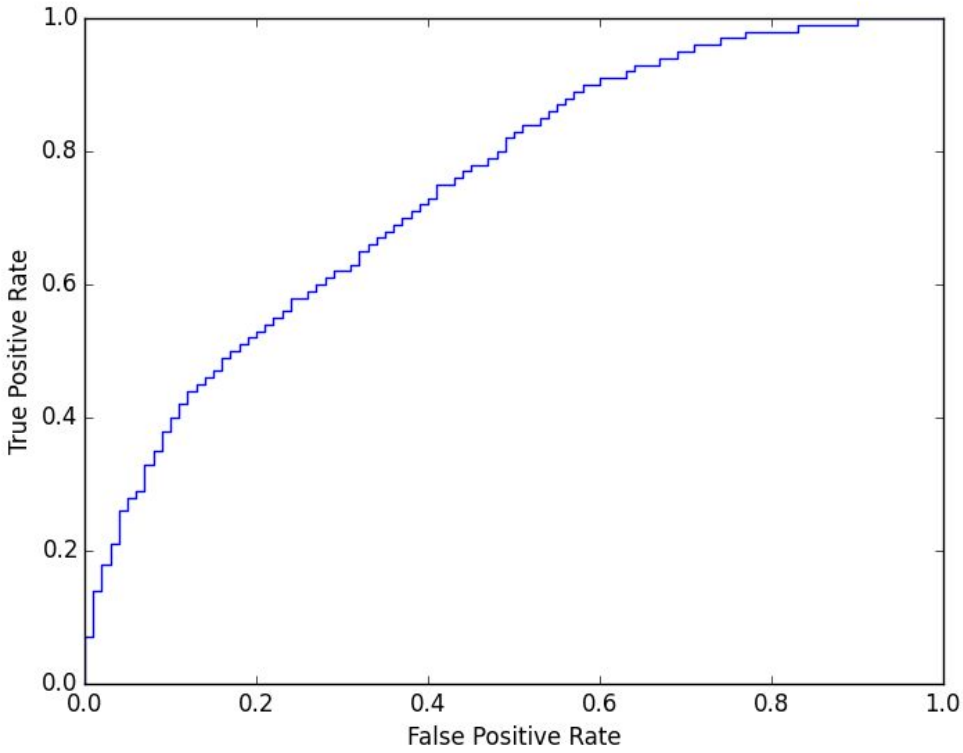
本实验中，我们一共有 6358 条蛋白质序列数据和 2716 对蛋白质相互作用的训练集，其中，阳性数据 1222 对，阴性数据 1494 对。我们知道，数据集越大时，所概括的蛋白质相互作用特性越多，预测就越准确。因此，将数据集中的全部阳性和阴性数据作为训练集，求出打分矩阵 M，然后用 M 矩阵对该训练集进行打分。作为对比，第二组则从数据集中的阳性和阴性数据部分各取出 300 对作为训练集。

因为对 C++不熟悉，所以程序是用 Python 写的，其中用到了开源的 numpy，matplotlib 等工具库。由于 Python 语言的特性使然，其代码简洁明了，短小精干，所以原本 1297 行的 C++代码用 Python 只需要 168 行就可以了。但是，Python 程序的性能较差，速度慢是众所周知的，所以，当要考虑到速度性能因素时，还是建议使用 C++。

第一组我们使用训练集的全部数据。程序运行结果如下图所示：

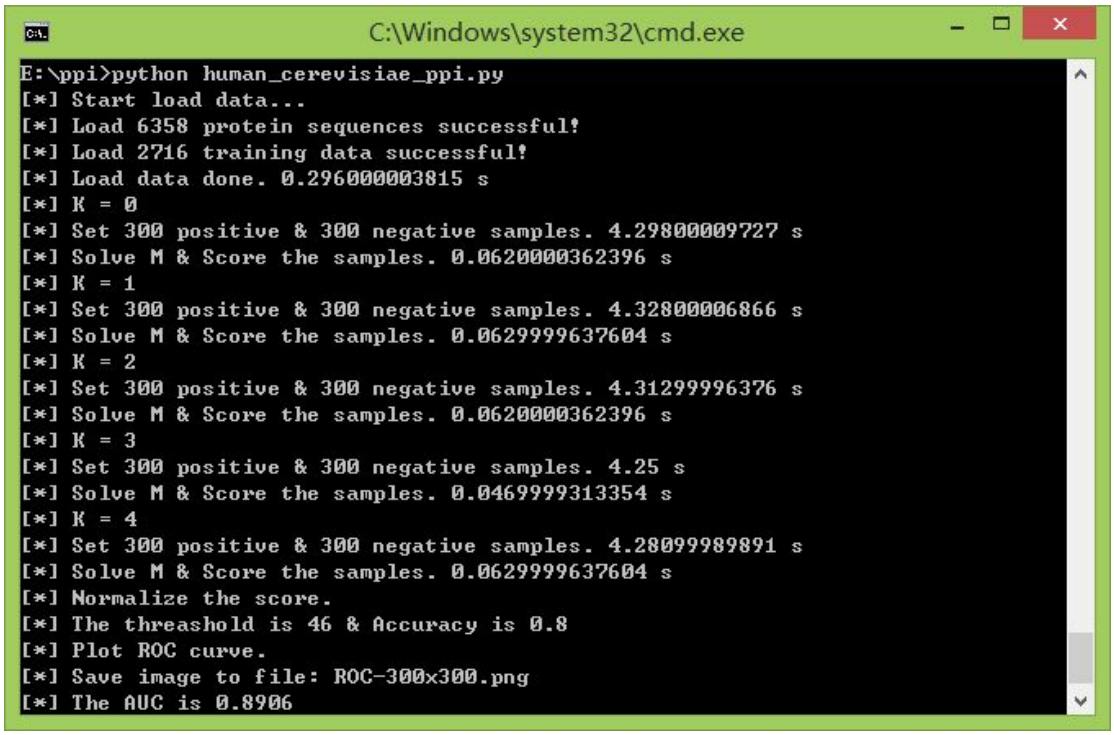


最后，我们得到的 threashold 值为 67，其对应的准确度为 68%。然后根据得到的结果，我们画出 ROC 曲线，ROC 曲线如下图所示：

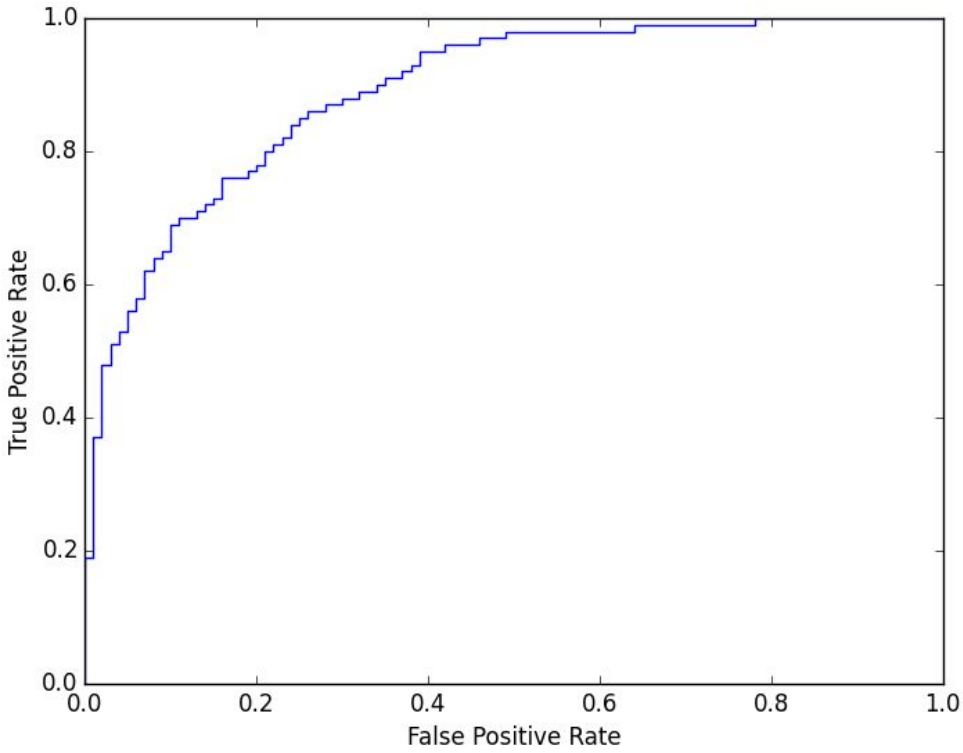


从 ROC 曲线中我们可以看出曲线整体比较接近左上角，表明结果的准确度较高。通过 AOC 曲线我们计算出其面积，即 AUC 值为 0.748，也就是说我们得到的打分矩阵 M 的效果还是比较好的。

作为对比，第二组我们只从中选取了阳性和阴性数据各 300 对。程序运行结果如下图所示：



ROC 曲线如下图所示：



最后得到的结果是，准确度为 80%，AUC 为 0.8906，会发现效果远超出第一组，主要原因是因为训练集和测试集数据量较小，更容易拟合，或是过拟合，所以其结果不具有参考性。

程序代码：

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

import math
import numpy as np
import matplotlib.pyplot as plt
from time import time

# Data files
PSQ_File = 'protein_sequence.txt'
TrainingData_File = 'training.txt'
sqMap = {}
positiveTrainingData = []
negativeTrainingData = []
pTrainingSample = 1222 # max: 1222
nTrainingSample = 1494 # max: 1494
Factor = [
    # Vector of hydrogen-bonding
    [0.0,0.13,52.0,49.5,3.38,1.43,49.7,0.35,1.48,1.58,3.53,1.67,49.9,1.66,0.0,2.10,51.
6,1.61,0.13,0.13],# Zimmerman_Score
    [8.1,4.9,10.5,11.3,11.6,5.7,13.0,5.2,5.5,8.0,10.5,9.2,12.3,8.6,9.0,5.4,10.4,6.2,5.2,5.
9],# Granthan_Score
    # Vector of secondary structure
    [0.66,0.59,0.95,1.01,1.56,0.60,1.46,0.60,1.19,1.52,0.98,1.43,0.74,0.96,1.56,0.96,0.
95,1.14,0.47,0.50],# Fasman_Score
    # Vector of Van der Waal's interaction
    [1.8,3.8,-4.5,-3.9,-3.5,1.9,-3.5,2.8,2.5,-1.6,-3.5,-0.8,-3.5,-0.7,-0.4,-0.9,-3.2,-1.3,4.
5,4.2],# KD_Score
    [8.1,4.9,10.5,11.3,11.6,5.7,13.0,5.2,5.5,8.0,10.5,9.2,12.3,8.6,9.0,5.4,10.4,6.2,5.2,5.
9]# BB_Score
]

print '[*] Start load data...'
startTime = time()
with open(PSQ_File) as f:
    acids = set('ALRKNMDFCPQSETGWHYIV')
    data = f.readlines()
    for i in xrange(0,len(data),2):
        ac = data[i].split('|')[1]
        sq = data[i+1].replace('\n','')
        if acids == set(sq):
            sqMap[ac] = sq
print '[*] Load ' + str(len(sqMap)) + ' protein sequences successful!'
```

```
with open(TrainingData_File) as f:
    for line in f.readlines():
        data = line.replace('\r\n','').replace('\n','').split(' ')
        if data[1] in sqMap and data[2] in sqMap:
            if data[0] == '+':
                positiveTrainingData.append(data[1:])
            else:
                negativeTrainingData.append(data[1:])
pNum = len(positiveTrainingData)
nNum = len(negativeTrainingData)
print '[*] Load ' + str(pNum + nNum) + ' training data successful!'

totalTime = time()-startTime
print '[*] Load data done. ' + str(totalTime) + ' s'

# 提取特征值
def getFactor(ac, k):
    acids = 'ALRKNMDFCPQSETGWHYIV'
    sq = sqMap[ac]
    f = Factor[k]
    return [f[acids.find(a)] for a in sq]

# 维度转换
def fourierTransform(arr, k=10):
    L = len(arr)
    C1, C2 = math.sqrt(2.0/L), math.pi/L
    return [C1 * sum([ arr[n]*math.cos( C2*(n+0.5)*(i+0.5) ) for n in range(L) ]) for
i in range(k)]

# 叉乘
def crossMulti(f1, f2):
    return [i*j for i in f1 for j in f2]

# 每一对蛋白质的特征向量相乘后的向量 X
def getX(sq1, sq2, k):
    f1 = fourierTransform(getFactor(sq1, k))
    f2 = fourierTransform(getFactor(sq2, k))
    X = crossMulti(f1, f2)
    return X

def setData(k=0):
    pos = [getX(d[0], d[1], k) for d in positiveTrainingData[:pTrainingSample]]
    neg = [getX(d[0], d[1], k) for d in negativeTrainingData[:nTrainingSample]]
```

```
    return pos, neg

# Fisher 判别法
def fisher(X1, X2):
    L1, L2 = len(X1), len(X2)
    M1 = np.sum(X1,0)/L1
    M2 = np.sum(X2,0)/L2
    S1 = 0
    S2 = 0
    for x in X1:
        tmp = x - M1
        tmp.shape = (1, M1.shape[0])
        S1 += tmp.T.dot(tmp)
    for x in X2:
        tmp = x-M2
        tmp.shape = (1, M2.shape[0])
        S2 += tmp.T.dot(tmp)
    Sw = S1 + S2
    W = np.linalg.inv(Sw).dot(M1-M2)
    return W

# 打分
def score(score, M, X, k):
    for i in range(len(score)):
        score[i, k] = M.dot(X[i])

def normalizeScore(scoreArr):
    norScore = scoreArr.copy()
    Lp, Ln = pTrainingSample, nTrainingSample
    m, n = norScore.shape
    for i in range(n):
        c1 = np.sum(norScore[:Lp,i]) / Lp
        c2 = np.sum(norScore[Lp:,i]) / Ln
        c = (c1+c2) / 2
        for j in range(m):
            norScore[j,i] = 100.0 / math.pi * math.atan(2*(norScore[j,i]-c)/(c1-c2)) +
50
    return norScore

def evaluate(score):
    maxRate = threashold = 0
    L, Lp = len(score), pTrainingSample
    for i in range(100):
        succ = sum(score[:Lp]>=i) + sum(score[Lp:]<i)
```

```
            rate = succ * 1.0 / L
            if rate > maxRate:
                maxRate, threashold = rate, i
        print '[*] The threashold is '+str(threashold)+' & Accuracy is '+ str(maxRate)

def ROC(score):
        FPR, TPR, FTPR = [], [], []
        P, N = pTrainingSample, nTrainingSample
        for threashold in score:
            FP = sum(score[P:] >= threashold)
            TP = sum(score[:P] >= threashold)
            FTPR.append( (round(FP * 1.0 / N, 2), round(TP * 1.0 / P, 2)) )
        FTPR = list(set(FTPR))
        FTPR.sort()
        FTPR = np.array(FTPR)
        plt.plot(FTPR[:,0],FTPR[:,1])
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        # plt.show()
        filename = 'ROC-'+str(pTrainingSample)+'x'+str(nTrainingSample)+'.png'
        plt.savefig(filename)
        print '[*] Save image to file: '+filename
        return FTPR[:,0], FTPR[:,1]

def AUC(x, y):
        direction = 1
        dx = np.diff(x)
        if np.any(dx < 0):
            if np.all(dx <= 0):
                direction = -1
            else:
                raise ValueError("Reordering is not turned on, and the x array is not
increasing: %s" % x)
        area = direction * np.trapz(y, x)
        return area

if __name__ == '__main__':
        LF=len(Factor)
        dataScore=np.zeros( (pTrainingSample + nTrainingSample,LF) )

        for k in range(LF):
            print '[*] K = '+str(k)
            startTime = time()
            pos,neg = setData(k)
```

```
            totalTime = time()-startTime
            print '[*] Set '+str(len(pos))+' positive & '+str(len(neg))+' negative samples. '+str(totalTime)+' s'
            startTime = time()
            M = fisher(pos, neg)
            score(dataScore, M, pos+neg, k)
            totalTime = time()-startTime
            print '[*] Solve M & Score the samples. '+str(totalTime)+' s'

        print '[*] Normalize the score.'
        norScore = normalizeScore(dataScore)
        finalScore = np.sum(norScore,1) / norScore.shape[1]

        evaluate(finalScore)
        print '[*] Plot ROC curve.'
        fpr, tpr = ROC(finalScore)
        print '[*] The AUC is ' + str(AUC(fpr, tpr))
```

三．在实习中收获最大与体会最深的内容

快乐的实习时光总是过得那么快，在这二周的实习期间，我学到了很多专业知识，更了解到了一些专业思想，让我真正体会到了做科研的每一步。我不仅学到了做科研的精神，同时也加深了对于本专业学习的浓厚兴趣。我很感谢刘老师和学长们的指导，他们都十分热情，教会了我许多知识。

所以，这次的实习让我明白了，做科研是一件十分严肃的事情，要有很认真的态度，我仍然需要不断的学习，需要更加努力奋斗，为今后的研究打下了良好的基础。

另外，在实习期间阅读了一些英文文献，收获了很多。一开始读英文文献大家都会感到十分头疼，但是随着慢慢看多了就好了。所以，这里总结了一些阅读英文文献的方法。

（1）从文献综述或综述类文献开始，概览全局，缩小范围。一般期刊论文中的摘要(Abstract)，统领全文，同时相当于阅读索引，告诉读者主要信息的分布情况，便于读者采撷自己需要的内容，一般会包括以下内容：

1. 领域近况（研究背景）
2. 主要课题/视角/研究方法（锁定研究范围）
3. 讨论的前提（保证论证的严密性，同时有自报家门的作用，因为不同的派别遵循的大前提多不同）
4. 研究目的（明确自己要达到的目标）
5. 篇章结构（明确研究步骤）

（2）阅读论文的过程逆向思考就是写论文的过程，互为逆过程的两件事如果能同时进行，一边读一边想，将事半功倍。

1. 首先，我们要告诉他们这是怎样的一个领域，主要研究的对象是什么，研究方法是什么，近些年取得了怎样的发展；简称为：做什么+怎么做+新鲜事

2. 其次，我们要告诉他们这样一个领域它的研究目的是什么；简称为：什么用

3. 再次，我们要告诉他们，我的研究在这样一个研究背景下着重那一部分，以及我选择这样一个研究重点有什么意义/原因，而研究的意义多跟研究现状的不足相关；简称为：着重点+为什么

4. 最后，我们要告诉他们，接下来我会跟大家以什么样的顺序讲哪些内容。让读者有一个鲜明的预期；简称为：列目录

（3）通过浏览标题，我们可以从一个大标题下小标题层级的多少来断定这部分信息的重要与否。并不是说论文中会放大量不重要的废话，但是为了保证行文的严密和逻辑的通顺，像定义 definition, 背景 background, 前提 premises, 预设 presumptions 这样的铺垫内容有其存在的必要性，而这些内容对于学科的内行来说是不需要每一次都仔仔细细看过去的。

（4）阅读文献的时候要不时地带入自己的知识体系，帮助甄别信息的优先级，同时也有助于理解性地记忆文献的大致结构和主要内容。最终达到在合上书本的时候，脑海中画树形图的境界。

还有一个的问题就是“怎样边读边想”？理论写作使用的英文通常大词长句连篇，生手不得不将注意力集中于读懂字面意思上，而无法像读中文理论那样随时连贯思考，导致读过和没读的感觉差不多。我的解决方法是：

1. 读完每一段后用中文完整翻译出最重要的一句话，如果实在把握不了，重点往往是首句或者末句。写在这一段旁边空白处。

2. 每读完一节，将所有翻译出的句子整理一下，寻找其中的思路和脉络，结合自己的想法写一段笔记。

3. 读完全文，将所有笔记复习一遍，找出其中不懂或者不连贯的地方，回去重读出问题的段落，修改笔记。

四．对实习工作的改进意见

总而言之，整个实习过程是很顺利很愉快的，也没什么问题。唯一的建议就是，实习地点离寝室较远，尤其是在天气炎热的暑假，每天跑来跑去都是一身汗。