## 用于生物信息自然语言处理的Python环境配置

1. **科学计算环境Python(x, y)**

<http://python-xy.github.io/downloads.html>中下载相应的release

在安装前移除任何其它Python distribution

1. **生物信息相关Package Biopython**

<http://biopython.org/wiki/Download>中下载对应的Biopython版本，Windows环境下推荐使用32位Windows Installer

在安装Biopython之前，先配置完毕Python科学计算环境，如Python(x, y)

* 1. **测试Biopython是否安装正确**
* **Biopython路径能否被Python正确识别**

运行Python.exe（或者Python(x, y)中的IDLE），启动Python console，输入以下语句

>>> from Bio.Seq import Seq

>>> from Bio.Alphabet.IUPAC import unambiguous\_dna

>>> new\_seq = Seq('GATCAGAAG', unambiguous\_dna)

>>> new\_seq[0:2]

Seq('GA', IUPACUnambiguousDNA())

>>> new\_seq.translate()

Seq('DQK', HasStopCodon(IUPACProtein(), '\*'))

* **其它测试**

由于采用Windows Installer安装后，documentation，tests，example code以及 source code都没有安装，因此需手动下载源程序代码包

启动命令提示符，进入到下载的源代码biopython-XXX根目录下，运行

python setup.py test

这等价于在Tests子目录下运行

python run\_tests.py

默认情况下将进行所有测试，如果某项单独测试没有通过，可以在Tests子目录下再次运行

python test\_XXXXX.py

会提供更多的提示信息

1. **安装C编译环境Mingw-w64**

Mingw-w64用于在Windows 64 & 32 bits环境中安装GCC编辑器，主页：<http://mingw-w64.org/doku.php>

在Downloads页面中，选择Mingw-builds

* **测试**

默认安装路径为C:\Program Files (x86)\mingw-w64\i686-6.2.0-posix-dwarf-rt\_v5-rev1\mingw32\bin，在该目录下运行

gcc –v

将显示gcc的版本等信息，如果看到此信息，表示安装成功

如有必要，将gcc的路径加入至系统环境变量Path中

1. **Object-Relational Mapping(ORM) Package peewee**

在<https://pypi.python.org/pypi/peewee>中下载源代码，运行进行安装

Python setup.py install

在Ubuntu下，需要先安装pip

apt-get update 更新

apt-get -y install python-pip 安装pip及所有必需Package

测试pip：pip -V

利用pip安装peewee：pip install peewee

* **测试**

进行测试

python setup.py test

等价于

python runtests.py

关于测试的详细说明在\peewee-2.8.5\playhouse\tests\README中。在Windows环境下，测试报错(与playhouse有关)，不影响使用

1. **自然语言处理相关Package** **Turku Event Extraction System (TEES)**

TEES是一个用于从生物医学文本中提取事件和关系的自然语言处理系统，主页：<https://github.com/jbjorne/TEES/wiki>，工作在一般**Unix/Linux**环境下

TEES的性能在BioNLP Shared Task中进行了评估(<http://www.bionlp-st.org/>)

TEES分析文本流程：

1. 使用GENIA Sentence Splitter (<http://www.nactem.ac.uk/y-matsu/geniass/>)将文本分割成语句
2. 使用BANNER (<http://banner.sourceforge.net/>)检测named entities
3. 使用基于David McClosky's biomodel的BILIP Parser (<http://bllip.cs.brown.edu/resources.shtml>，Stanford格式版本：<http://nlp.stanford.edu/software/lex-parser.shtml>)来解析文本
   1. **安装**

下载源代码后，可运行

(sudo) python setup.py install

将其安装成一个模块，也可以不进行安装，而直接解压缩后使用

* 1. **配置**

运行

python configure.py

进行配置。在配置过程中，会提示需要安装Ruby编程语言(sudo apt-get install ruby)，测试(ruby -v)；安装Abobe Flex编程语言(sudo apt-get install flex)；安装Java编程语言(sudo apt-get install default-jre; sudo apt-get install default-jdk)，测试(java -version)

* 1. **测试**

测试TEES在BioNLP 2011 GENIA建模数据集上的性能（-o表示结果保存时的前缀；-i表示输入，可以为XML文件、TXT ASCII文档或者含有多个文档的压缩文件；-m表示使用的模型，含有“-devel”和“-test”两种后缀，当没有后缀时，默认为“-test”后缀，推荐使用“GE11-test”模型。如果使用相对路径的话，就在调用classify.py函数的路径下）

python classify.py -m GE11-devel -i GE11-devel -o OUTSTEM

测试解析PubMed上一篇id为9668063的文章摘要（会联网下载）

python classify.py -m GE11 -i 9668063 -o OUTSTEM

报错，查原因为安装Stanford Parser后系统环境变量没有更新，重装该模块后问题解决，正常应为

STANFORD\_PARSER\_DIR = '/home/hill103/.tees/tools/stanford-parser-2012-03-09'

结果在OUTSTEM-pred.xml.gz中

* 1. **批处理**
     1. **批处理配置**

**安装Psyco模块**

Psyco是一个提高Python执行效率的解释器，已停止维护，可采用PyPy代替

**安装SLURM**

**文件准备**

批处理多个文本，压缩包中需要包含文件夹，文件夹中才是各个文本文件。每个文本文件中，如果包含多行，各行的文本都会进行解析

文件树

dir1

|- input1.tar.gz

|- otherfile.txt

|- dir2

|-input2.tar.gz

|-input3.tar.gz

**命令示例**

python batch.py -i /dir1 -r '.\*input[1-3]\*.tar.gz$' -n SLURM --limit 100 -c "python /ABSPATH\_TO\_TEES/classify.py -i %a -o %a -m GE"

或者

python batch.py -i /media/hill103/12F03AD9F03AC2B1/test -r '.\*input[1-2]\*.tar.gz$' -n SLURM --limit 100 -c "python /media/hill103/12F03AD9F03AC2B1/my file/jbjorne-TEES-1125ab0/classify.py -i %a -o %a -m GE"

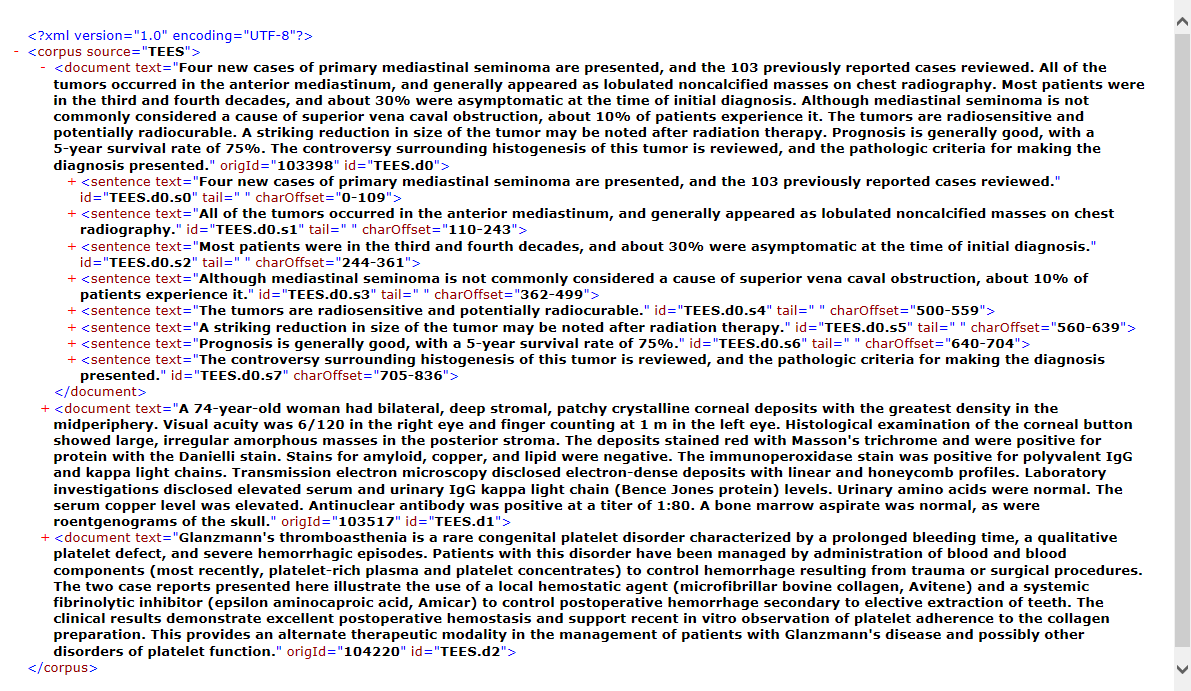
* 1. **结果解析**

在结果XML文件中，最外层tag为**document**，对应于输入的每一个文本文件，包含3个属性：

**id** = “TEES.dX”，X为从0开始取值的整数；

**origId**，对应于原始的文本文件名；

**text**，对应于原始的文本文件中内容。



每一个document中，包含多个sentence tag，对应于文本中的多个语句，每一条语句都会进行分析，其中包含属性：

**charOffset** = “XX-XX”，对应于该语句在原文本中的位置，起始点包括第一个数字，终止点不包括最后一个数字；

**id** = “TEES.dX.sY”，X跟随document中的id，Y为从0开始取值的整数；

**text**，对应于每条语句的内容

每一个sentence tag下面，肯定具有analyses tag，但这不是关注重点。如果解析该语句有发现entity或者interaction，会出现对应的tag，否则没有。

**entity** tag重要属性包括：

**id** = “TEES.dX.sY.eZ”，X跟随document中的id，Y跟随sentence中的id，Z为从0开始取值的整数；

**text**，对应于该entity的文本内容；

**type**，该entity的类别（基因名称也会归为蛋白质类别，因此该信息可不记录）；

**event** = "True"，说明该entity是一个生物事件，会有一个对应的interaction。

**interaction** tag重要属性包括：

**e1** = "TEES.dX.sY.eZ"，对应发现生物事件的entity（语句）；

**e2** = "TEES.dX.sY.eZ"，对应该事件对应的entity（一个基因或蛋白质）。



综上所述，分析流程为：寻找每一个**sentence**下面是否存在**interaction** tag。如果存在，查询**e1** tag所在entity的**type**，以及**e2** tag所在entity的**text**和**type**，只保存这些信息即可。