水稻基因与性状的共句显示

赵柯韦 [1], 黄奇楠 [2]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国
[2] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

为了挖掘和发现水稻基因和性状之间的联系，以及在先前的实验中对这一关系的研究。我们尝试从和水稻相关的文献中提取出与水稻性状和基因相关的信息和数据，并且通过网络建模、依存树和词云等方法对收集和整理到的数据进行分析并从其中提取出具有一定价值和意义的信息。

关键词: 水稻，基因，性状，共句显示，词云，依存树，网络

1 课题概况

本项目难度适中，而且涉及我们相对较为感兴趣的水稻的研究领域，在协商之下我们选择了这个题目来撰写课程论文。本文致力于挖掘过去多年研究中所涉及的对于水稻基因与性状的关系信息，预计可以基于挖掘出来的信息去构建一个“性状–基因”和多对多网络。

2 数据

本项目的文献原始数据是通过 Pubmed 所提供的 edirect 工具检索并下载所有与水稻 (Oryza sativa L.) 相关的英文文献摘要，并保存为 json 格式。将 PMID 从原始数据中提取出来，并保存为 txt 文件，将该文本文件作为 Pubtator 脚本的输入，批量下载 Pubtator 数据并保存为 txt 格式。

本项目的性状原始数据由老师所提供的 RTO 形状本体数据提供，下载后保存为 txt 格式。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

在本项目中所主要使用的算法是依存树分析算法，该算法会对我们提取出的语句进行分词，并且根据分词结果和训练结果为该语句构建依存树，依存树中的每个节点代表着一个词或者说是句子组分，而依存树中的边则表示着词和词之间的依存关系（控制词，依存词）。这种分析方式有利于我们提取句子主干和明确基因和形状在这个语句中的性质和交互关系。

实现相同的功能我们同样可以使用 CRF 或者逻辑回归去实现类似的功能。但是，在该项目中这两种算法我们认为都不如依存树算法合适，首先，逻辑回归算法和 CRF 所需要的训练数据需要手动注释，而受限于我们的知识水平和所接触的数据广度，我们很难做出精确且有价值的训练数据集。其次，逻辑回归和 CRF 的算法核心和依存树不同，二者本质上都是一种分类算法，并不会生成词和词之间的关系，而依存树所构建的依存关系和我们在实验中所需求的“性状–基因”关系相对更加契合

3.2 研究方法中的核心思路

在进行该项目时，我们的核心思路大体分为三个部分：数据的下载和整理-> 数据的提取-> 数据的分析与可视化。在数据的下载和整理的过程中尽可能多地同时获得相关的注释信息，并且将数据整理成相对更加有利于提取和分类的形式防止因为数据形式的问题导致数据提取的难度提高。在数据提取的过程中，我们需要尽可能的考虑到基因和性状所在语句中所可能出现的各种情况，并尽可能地覆盖所有结果。得到数据后，需要对数据进行分析和判断，这一过程中，我们应尽可能多的应用课堂上所学习的各种分析手法，多角度，多层次地体现结果的特征和性质。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

本项目中的方法部分大多数都参考自课堂上讲授的内容，其中在分句和依存树分析部分基于本项目具体情况做出了部分调整：

首先，由于 Pubtator 所给出的注释中包含了 GO 的 start 和 end，所以我们并没有直接使用相关模块中的分句语句，而是自己编写了脚本将标题和摘要合并并记录每一个句子（标题视为首句）的 start 和 end，用于方便后续的语句提取。

其次，根据依存树的结果，我们发现基因和性状之间的联系不能简单的使用课堂上所讲授的 nsubj-ROOT-dobj 这种简单的模型来反映。基因在很多情况下是作为语句中的同位语或者其它成分存在，除此之外，形状本体很多情况下是一个短语而非一个独立的单词，所以，我们在进行依存树分析的时候也进行了相关的调整来适应这个实验：将过滤条件中引入了除 nsuj 和 dobj 之外的 nsubjpass 来尽可能捕获更多的符合条件的语句；如上文所述，我们采取了将词组转化为单词来防止依存树分析将词组分解拆分，来实现整体的标注。

在词云的绘制中，不同于使用 R 语言处理数据并绘制图像。由于 R 语言包 wordcloud2 存在缺陷，我们无法使用这个包去绘制指定图形的词云。我们采取了使用 R 语言进行语料库的过滤、分析以及抽取高频率的词汇表。基于 R 语言的处理结果，再调用 python 中的 jieba 模块实现分词吗，并通过 wordcloud 包对分词的结果进行词云的绘制，最终得到了如本文中图 1 中所示的特定形状的词云。

4 算法实践和代码编写要求

4.1 任务描述

从原始文献摘要中提取同时含有 GO 和 TO 的语句，筛选出这些语句中可以体现明确的性状和基因的关系的语句。并对这些语句进行分析，将分析结果可视化，同时从中获得相关的生物学意义。

4.2 实验设计

共句表达的提取：由于我们所使用的数据来源于 pubtator，所以先天带有和基因相关的注释，我们首先删除了所有非基因注释而且合并了标题和摘要。其次，根据上文所述的分句方法（生成每句话的 start 和 end）可以很方便的提取出含有 GO 的语句。另一边，我们提取了 RTO 文件的 name 项和 synonym 项整理为一个 TO 字典。通过调用 re 包中的相关语句实现正则表达和匹配，并通过循环使字典中的每一项都和每一句含 GO 的语句进行匹配，并根据匹配结果筛选出同时含有 GO 和 TO 的语句。

依存树筛选：为了确定这些语句中基因和性状之间是存在某种联系，而非只是单纯的，恰好的出现在同一个语句中，我们对这些语句进行了依存树分析，并保存了依存树分析的完整结果。根据这些结果去判断删除还是保留这个句子，该筛选过程先由脚本进行较为严格的筛选，在删除的句子中部分由人工观察保留下来。

分析与可视化：将筛选出来的句子整合成为一个语料库，在进行数据清洗后建立一个词云去反应各种词语在这些语句中出现的频率。另一方面，通过 TO 标准字典将所有的 synonym 都转化回 name，然后基于 GO-TO 这一对关系去构建一个关系网络，并且通过调用相关的模块，计算该网络的包括中心性，连通性在内的一系列参数来反映该网络的性质。

其中较为复杂的是共句表达的提取，因为我们事先无法确定 TO 在句子中的形式以及出现的位置，所以我们只能对句子和 TOname 都进行了格式统一化，即删除所有标点符号同时将所有大写转化为小写后进行匹配。但这种相对粗暴的操作也带来了一系列问题，首先，有些对大小写有严格要求的 TO 被错误匹配到了小写的一般单词上，如 BY 和 An 就会出现这种问题。其次通过阅读 RTO 文档，我们发现了许多同义词或者说字典中的许多词是全集与子集的关系，比如，某句中出现了 molecular hydrogen sensitivity，在进行正则匹配的时候 hydrogen sensitivity 和 molecular hydrogen sensitivity 这两个对象都可以匹配，但是实际上应该只允许 molecular hydrogen sensitivity 正确匹配到该句。

除此之外，在依存树分析中，也出现了一系列未曾设想的问题。首先，由于 TO 往往是以词组的形式存在，在进行依存树分析的时候，依存树往往会将这个词组拆分然后分别标注。另外，通过观察未经处理的依存树结果，我们也发现了在部分语句中，依存树不能很好的识别基因实体，这导致了标注缺失的问题。我们尝试采用了代换的方法来解决这两个问题，即使用 TO1，TO2，TO3 · · · · · · 来代替语句中出现的所有 TO，使用 GO1，GO2，GO3 · · · · · · 来代替所有出现在语句中的 GO，以便于计算机有效识别。

基于获取的共句信息，我们调用 networkx 模块构建了一个性状与基因互作的网络，并将关系所出现的频次作为参数赋值给了代表该关系的边。基于研究的问题，我们选择了点度中心性和度分布这两个参数作为反应该图性质的主要参数。并且我们通过 Cytoscape 软件实现了该网络的可视化。

4.3 实验关键代码

```
#代码来源：赵科韦编写
#GO提取与文本分句
for row in f:
    if '|t|' in row:
        key = row.split('|t|')[0]
        title = row.split('|t|')[1].strip()+'␣'
        p_d[key] = {}
        p_d[key]['paper'] = title
        p_d[key]['annotation'] = []
    elif '|a|' in row:
        key = row.split('|a|')[0]
        abstract = row.split('|a|')[1].strip()+'␣'
        p_d[key]['paper'] = p_d[key]['paper'] + abstract
        start = [0]
        stop = []
        for i in range(len(p_d[key]['paper'])):
            if p_d[key]['paper'][i:i+2] == '.␣':
                stop.append(i+2)
                start.append(i+2)
        del start[-1]
        sentence = []
        for i in range(len(start)):
            sentence.append([start[i],stop[i]])
        p_d[key]['sentence'] = sentence
    elif row != '\n':
        note = row.strip().split('\t')
        if note[4] == 'Gene':
            p_d[key]['annotation'].append(note)
```

```
#代码来源：赵科韦编写
```

```
#TO匹配与过滤
for TO in dictionary:
    if TO not in special_TO:
        pure_TO = TO.translate(str.maketrans('', '', string.punctuation)).lower()
        pattern = re.compile('\\b' + pure_TO + '\\b')
        m = re.search(pattern,pure_sentence)
        if m != None:
            gts[key]['TO'].append(TO)
    else:
        pure_sentence = key.translate(str.maketrans('', '', string.punctuation))
        pure_TO = TO.translate(str.maketrans('', '', string.punctuation))
        pattern = re.compile('\\b' + pure_TO + '\\b')
        m = re.search(pattern,pure_sentence)
        if m != None:
            gts[key]['TO'].append(TO)
```

```
#代码来源：赵科韦编写
#依存分析
nlp = spacy.load('en_core_web_sm')
for key in gts.keys():
    sentence = key
    num =1
    for i in gts[key]['GO']:
        sentence = sentence.replace(i,'GO'+str(num))
    num = 1
    for i in gts[key]['TO']:
        sentence = sentence.replace(i,'TO'+str(num))
    word_pos_dict = {}
    doc = nlp(sentence)
    for token in doc:
        word_pos_dict[token.dep_] = token.text
    gts[key]['depency_result'] = word_pos_dict
```

5 主要的生物信息学实验和实验结论

5.1 依存关系分析结果

依存关系分析的结果较为不理想，通过观察依存树分析结果我们推测这种情况可能是由三方面原因所引起的：一、语句来源于科研论文，其语句句法的复杂性相对较高，依存关系的判断本来就难度较高；二、我们在实验中所使用的训练数据可能不能很好地处理生物文本信息，导致了在进行依存分析时，许多 TO 和 GO 出现了无法识别的情况；三、我们所使用的句子中标点符号数量和种类都偏多，这可能也是导致较差结果的原因。但是基于依存分析结果我们仍然获得了 41 个具有强烈相关关系（即作为主语和宾语）的性状基因对。分析这些关系可以看出，对于水稻的基因与形状的研究主要还是立足于农业的，并且主要集中在水稻的抗性和生产这两个方面，即水稻的不同基因可以带来怎样的有利于农业生产的性状，在我们的提取出的结果中涉及了干旱耐性，高盐耐性，一些氨基酸在水稻中的含量，水稻的生长发育等一系列性状。

5.2 网络模型可视化

从网络模型中观察可以看出，共句显示结果中所涉及的基因本体和性状本体数量较多，且大部分性状、基因可以通过共句关系构建一个规模相对较大的网络，只有少部分基因和性状是游离在这个大规模网络之外的子网。我们分析认为可能会有三种原因导致这种情况：1）确实存在部分性状和基因间存在较为独立的联系。2）此网络是基于过往的研究成果进行的信息挖掘，可能这些游离的网络与大型网络存在某种联系，

但现在未被发现或者没有在这些摘要中体现。3）pubtator 无法完全提取所有的 GO 数据，且 TO 源文件和匹配过程都存在遗漏的可能性。因此导致了部分关系没有被我们挖掘出来。

5.3 网络度分布可视化

根据对网络的度分布结果的假设检验和可视化可以看出，该网络的度分布与幂律分布较为相近，即少数节点拥有大量连接，而大多数节点只拥有少数连接，具有着严重的不均匀分布性 [1]。也就是说少数的 TO 和大量的 GO 存在联系，少数 GO 与大量的 TO 存在联系。这在一定程度上体现了在水稻中存在部分较为关键的基因这些基因可能同时调控着下游大量的性状，同时也存在着部分调控机制或作用机制较为复杂的性状，这些性状受到大量基因的调控。该网络同时具有鲁棒性和脆弱性，一方面对于随即故障的容错能力强，另一方面，如果关键节点受损，很可能破坏整个网路的功能。对于水稻这个系统来说，一方面其作为一个生物系统拥有一定的抗逆能力和适应能力，而另一方面如果水稻的关键基因，或者关键功能（如光合作用或者糖酵解过程）收到损伤，有可能会导致水稻的生长不良甚至死亡。

5.4 实验结果展示

以下是本次课程项目结果的部分图片展示 [1]：

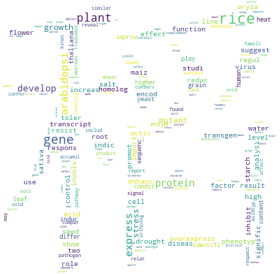

图 1: 水稻词云图

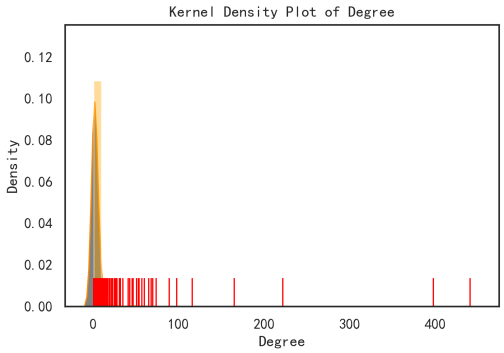

图 2: 网络度分布密度曲线图

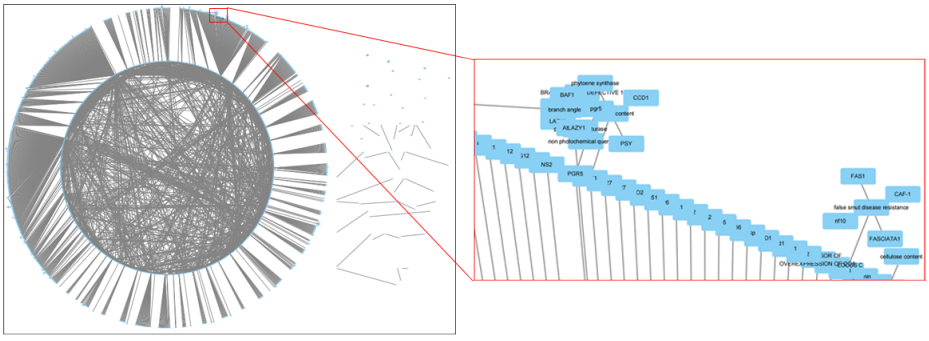

图 3: 形状本体与基因本体的共句关系网络

[1]图 1 在 R 中运用 wordcloud2 包实现绘制；图 2 在 python 中运用 matplotlib.pyplot 包绘制；图 3 利用 Cytoscape 绘制而成，再局部放大

本次实验除了利用图像对数据进行可视化外，我们还将网络度分布和词频进行了表格可视化处理，具体结果如下：

表 1: 网络度分布表格 (部分)

TO	GO	Freq
growth	mL-1	2
phenotype	pex5	1
heat	AtSIZ1	3
morphology	ROXY1/2	1
rat damage resistance	Exp.1	2

表 2: 词云频次表格 (部分)

Word	Freq
rice	908
gene	677
plant	556
express	487
arabidopsi	474
protein	413
develop	345
growth	318
stress	313

6 后记

6.1 课程论文构思和撰写过程

本项目主要侧重于数据的处理和分析上，在算法部分偏向于薄弱，因此在撰写论文时，我们也主要侧重于讲解我们在数据处理和在结果分析上所遇到的问题和经验。并且尽可能地使用更多的分析手法来从不同层次和不同的方向分析我们所获得的结果，同时对这些分析结果也尽可能地做出可视化和再分析。并根据这些得到的数据，寻找到其中相关的生物学知识和解释。

在论文结构上，我们并没有严格遵循我们进行实验时的时间顺序，而是根据本项目的逻辑关系，将各个步骤进行了整合和归类，然后，进行了论文的撰写。

6.2 所参考主要资源

本文所使用的性状本体信息来源于由夏静波老师所提供的 RTO 数据 [2]，所使用的摘要数据信息来源于由 Pubtator 下载的信息。在自然语言处理中所应用的主要知识和思路主要参考自夏老师上课时所使用的 PPT 和课堂教学内容 [3],[4]。在网络构建和分析的部分主要参考了马彬广老师在系统与合成生物学上所使用的 PPT 和教学内容。

6.3 代码撰写的构思和体会

在最初的代码思路中，我们本来是想要基于 NCBI 数据库下载所有关于水稻的基因，并进行手动匹配的。不过在后续的处理中，我们发现这样的做法容易存在大量的错配和未匹配。于是，在 GO 的提取上我们直接使用了 Pubtator 的注释结果，并改变了原先的分句方法以适配 Pubtator 的注释格式。

在 TO 的匹配上，由于我们抹除了语句和 TO 的所有格式和标点，这使得结果中出现了两个难以解释的峰值数据 “BY” 和 “An”，我们在之后选择对这部分的 TO 做特殊处理，即严格匹配大小写和格式。除此之外，由于我们之前的数据储存格式是 sentence+TO+GO 这种格式，其中记录了大量无用且重复的信息，之后我们将数据储存格式改为 sentence+TO_set+GO_set，有效节省了储存空间，也方便了后续的过滤。

在共句的过滤中，我们主要针对三种情况进行过滤：1）包含型：在同一句中，一个 GO 被另一个 GO 完全包含。2）特殊型：如上文所述，对特殊型 TO 进行更加严格的匹配。3）空值型：在进行完上两步过滤之后，清除含有空列表的项目。本来在计划中还存在着依存信息过滤，但是由于上文中所示的原因，该部分并未在本文中应用和展示。

在网络搭建中，我们认为一个 TO 项目的 name 和 synonym 不应该在网络中处于两个不同的节点，所以我们调整了对于 TO 的处理，让其不仅产生一个 TO 字典，同时也会产生一个标准字典，即 name/synonym :name。来实现输入后返回该输入的标准命名。

本项目的代码难度偏低，但涉及问题和细节较多，不仅要求我们在初次编写代码时就尽可能思考全面，而且需要我们在撰写代码时不断改进原先的代码以满足当前发现的问题。

6.4 生物信息学实验设计的构思和体会

根据项目要求，我们在初步构思该实验的时候仅仅考虑到了数据的提取和共句信息的提取。但是，随着学习的深入，我们逐渐意识到仅仅获取结果并不能算是一个完整的研究过程，这个结果如果不进行分析和处理，我们是很难从结果中提取出具有生物学意义的结论。在思考如何分析数据时，一方面，我们认为在本课程上所学习的词云方法是一种直观且具体的方法去反应这些句子的高频词汇；另一方面，共句表达本质上是一种本体与本体之间的联系，所以我们认为针对这些共句关系构建一个网络，并且将关系出现的频率作为属性赋值给网络是一种很好的数据处理方法。

我们组的实验设计是随着实验的推进而不断发生改变的，有一些新内容（如词云、网络）被添加到实验设计中，也有一些内容（依存树）由于结果较差，虽然仍然进行了实验，但没有被包含在结果之中。这些缺陷和后期完善的部分将会成为我们未来设计实验时的教训和经验。

6.5 人员分工

在本项目中，赵柯韦和黄奇楠采取了独立编写的同时相互交流和参考的方式进行，两人均以不同的方式实现了项目目标。论文撰写也由两个人共同完成。

赵柯韦：实验设计；代码编写；论文的内容书写，论文核查与修改。

黄奇楠：实验设计；代码编写；论文的格式书写，论文核查与修改。

7 附录

S1. 2021 年课程网页. `https://hzaubionlp.com/course-bionlp-and-kd/`

S2. 课程论文可以参考使用的基础代码，可参考 GitHub 页面, `https://github.com/bionlp-hzau/`

S3. 课程论文所使用的全部代码、数据及详细结果，可参考 GitHub 页面，`https://github.com/Allen-ZKW/NLP_HZAU/tree/term`

参考文献

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[2] Xinzhi Yao, Jingbo Xia, Kaiyin Zhou. Tomapping. `https://https://github.com/bionlp-hzau/TOMapping/`.

[3] Jingbo Xia. Tutorial4wordcloud-basic. `https://https://github.com/bionlp-hzau/Tutorial4WordCloud-Basic/`.

[4] Kaiyin Zhou, Jingbo Xia. tutorial for dependency tree. `https://github.com/bionlp-hzau/tutorial4dependencytree/`.