

lazybios

首页 / 书单 / 关于

D3.js数据可视化使用小记

最近毕设基本告一段落，题目是数据可视化相关的项目，用到了d3.js,这里集中回顾记录下遇到的一些问题。

毕设要求基于Web显示柑橘蛋白质分支间相互做用的关系，蛋白质总数共8195个，共有12491组不同关系。当然这些蛋白质的检测工作与我无关，数据是现成的，我的任务是拿到它们并把它们之间的关系直观的显示出来。

所给数据是csv格式的，而且信息是分散到多个不同的csv文件中的，考虑到最终的显示的是网络图，所以我决定将csv转换成关系型数据存储，最后根据查询条件动态生成自定义符合d3.js force graph 要求的json格式。基于上面的需求，我的实施方案如下：

前端 JQuery D3.js Bootstrap 后端 Python Tornado Sqlite3(后边会转移到Mysql上)

Step 1 格式化csv文件，清洗数据

后端语言用的Python，当然脚本也用Python写的，用到了csv,sqlite3模块。处理文件较大约30MB的样子，电脑比较渣，耗时大概3个小时才建起数据库 囧...

Step 2 定义Json生成接口

按照D3里Force Layout所要求的JSON格式做定义，因为仅仅是一对多的关系，所以仍然是nodes与links结构，只是多了些附加字段用来做其它功能的扩展。比如 name,exp,summ,score1,score2 等。

```
{
  "nodes": [
    {
      "summ": {
        "bin": "30.7",
        "mapman": "signalling.14-3-3 proteins",
        "annotation": "annotation"
      }
    }
  ]
}
```

```

    },
    "name": "Cs2g04850.2",
    "exp": {
      "flower": 95.2894,
      "leaf": 65.7036,
      "fruit": 34.7638,
      "mixture1": 64.9308,
      "mixture2": 40.8747,
      "mixture3": 56.1705,
      "callus": 17.1658
    },
    "center": "y"
  },
  ...
],
"links": [
  {
    "source": "Cs2g04850.2",
    "target": "Cs6g10590.1",
    "score2": 1,
    "score1": 0
  },
  ...
]
}

```

Step 3 D3渲染可视化效果

这是本次毕设的核心部分，除了D3还用到了JQuery，Ajax技术，JQuery与Ajax主要做为前端交换使用，用于提交查询，异步获取Json数据。当然这里面也遇到不少问题。

- 中心点(即一对多的1)无法定位，每次生成都是随机定位

解决办法: node中有一个fixed属性,取值为boolean值，true代表位置固定，false代表不固定，在遍历所有nodes时候，判断出中心点，为其设置固定坐标(x,y)，同时设置fixed属性即可

- Firefox使用表单通过异步请求json数据会无法返回，同样的在chrome中就不会有问题

解决办法: 这是个浏览器兼容问题，之所以firefox中会有问题，因为Firefox中form表单的默认行为，影响了 d3.json 方法的异步请求，同样的使用JQuery的 jquery.getjson() 也会有问题。有两个方案可行，一是不要用表单提交，直接用 input ，或者改变表单的默认行为，后者比较麻烦。另外动态生成的json数据要设置 Content-Type 为 application/json

- 缩放与平移

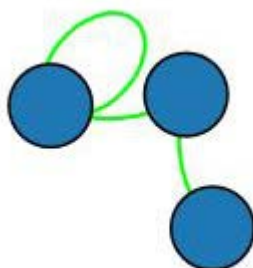
解决办法: 对svg容器添加 `d3.behavior.zoom()` 行为, 并设置zoom事件监听器。最关键的地方是要记得, 在svg后追加一个 `g` 容器, 代码如下

```
var vis = d3.select("body")
    .append("svg:svg")
    .attr("class", "stage")
    .attr("width", w)
    .attr("height", h)
    .attr("pointer-events", "all")
    .append('svg:g')
    .call(d3.behavior.zoom().on("zoom", zoom))
    .append('svg:g');

vis.append("svg:rect")
    .attr("class", "overlay")
    .attr("width", w)
    .attr("height", h);

function zoom() {
    vis.attr("transform", "translate(" + d3.event.translate + ")scale(" + d3.event.scale + ")");
}
```

- 自我指向连接设置, 如下图



解决办法: 修改json中links的映射关系, `source`与`target`分别引用到nodes中的点, 映射函数用的是 `d3.map()`, 因为中心点的关系在nodes中用两个, 所以使用 `d3.set()` 建立nodes的map结构时会发生覆盖问题, 导致后面建立link关系时会出现孤立的点, 这时就需要在建立map时, 对map中是否已经存在该键做判断, 代码如下

```
function mapNodes(nodes){
    nodesMap = d3.map()
    nodes.forEach(function(n){
```

```
        if (!nodesMap.has(n.name)){
            nodesMap.set(n.name,n)
        }
    });
    return nodesMap;
}

nodesMap = mapNodes(ppi_json.nodes);
ppi_json.links.forEach(function(l){
    l.source = nodesMap.get(l.source)
    l.target = nodesMap.get(l.target)
});
```

SVG, PATH用法可以参考mozilla的[MDN](#)

总结

以上是这次毕设的总体思路与一些自认为关键的细节点，因为之前不是专门做js的，所以有罗嗦了一点，总的来说这个项目还是很锻炼人的，让我又把web相关的东西（前后台）理了一遍，后续我会把这个项目搭建到SAE上，并把源码放到github上



日拱一卒 · 功不唐捐

可以十年不将军 不可一日不拱卒

日拱一卒

Made with Jekyll using Sparrow theme



支持翻译PDF和WORD文档

支持将英文翻译成中文

要获取最佳完整体验，请将浏览器升级到最新版

上传文档 关闭