

- guluT20141102
- 如发现翻译不当或有其他问题可以通过以下方式联系译者:
- 邮箱: [zhang\\_tianxu@sina.com](mailto:zhang_tianxu@sina.com)
- QQ群: [D3数据可视化](#)205076374, [大数据可视化](#)436442115

捆绑布局实现了Danny Holten的分层边缘捆绑式算法。对于每个输入的连接, 都计算穿过树的路径, 父层到最小的共同祖先, 然后退回到目标节点。这些节点随后被用来和其他的分层布局一起使用, 例如簇图用来生成节点之间的捆绑样条线。例如, 看这个软件依赖关系的可视化。

## d3.layout.bundle()

构造一个新的默认的捆绑图。当前, 捆绑图是无状态的, 因此, 只有一个默认的配置。返回的布局对象是一个对象也是一个函数。这样, 你就可以像其他函数一样调用布局, 布局有额外的方法改变他自己的行为。像D3中的其他类一样, 布局遵循链式语法, setter方法返回布局自身, 允许在一个简单的申明中调用多个setter方法。

## bundle(links)

- parent - the parent node. 通过指定的links数组计算捆布局, 返回通过最小共同祖先计算的路径(从来源到目标)。
- source - 源节点。 • target - 目标节点。 此外, 每个节点都必须有一个属性: • parent - 父节点。

这是层布局生成域的一个子集。布局返回的值是一组路径, 其中每条路径代表一组节点。这样, 捆绑布局不会直接依据样条线计算。而是返回一个节点数组明确地表示样条线的控制点。你可以使用这个数据和d3.svg.line 或者 d3.svg.line.radial一起使用, 生成样条线自己。例如, 如果你要使用一个簇, 可以这样写: `var cluster = d3.layout.cluster().size([2 * Math.PI, 500]);`

层次边捆绑的合适的线生成器可能是: `var line = d3.svg.line.radial().interpolate("bundle").tension(.85).radius(function(d) { return d.y; }).angle(function(d) { return d.x; });`

捆绑布局是设计来和线生成器的“捆绑”插值器模块一起使用的, 虽然从技术上讲, 你可以使用任何插值器或形状生成器。Holten的捆绑强度参数都会暴露为线的张力。 bundle 部分官方API:

<https://github.com/mbostock/d3/wiki/Bundle-Layout>。

咕噜 译 2014-11-30 09:18:08