

- [\[\[Bundle|捆绑布局\]\]](#) - 对边使用Holten 层次捆绑算法。
- [\[\[Chord|弦布局\]\]](#) - 从关系矩阵生成一个弦图。
- [\[\[Cluster|簇布局\]\]](#) - 将实体聚集成树状图。
- [\[\[Force|力布局\]\]](#) - 模拟物理力排放相连节点的位置。
- [\[\[Hierarchy|层次布局\]\]](#) - 这是个抽象布局，可派生一个定制的层次布局。
- [\[\[Histogram|直方图布局\]\]](#) - 使用量化的箱计算数据的分布。
- [\[\[Pack|包布局\]\]](#) - 用递归的圆形包装生成一个层次布局。
- [\[\[Partition|分区布局\]\]](#) - 递归地将节点树分割为旭日状或者冰柱状。
- [\[\[Pie|饼布局\]\]](#) - 计算饼图或圆环图中弧的开始和结束角度。
- [\[\[Stack|堆叠布局\]\]](#) - 计算堆叠图或者面积图的基线。
- [\[\[Tree|树布局\]\]](#) - 整齐地排列树节点。注意簇布局不是整齐的。
- [\[\[Treemap|矩形树布局\]\]](#) - 使用空间递归分区算法展示树的节点。