

```

const configs = {
  container: document.getElementById('main'),
  nodeLimit: 1e5,
  linkLimit: 1e7,
  node: {
    strokeWidth: 0
  },
  link: {
    strokeWidth: 0.5
  }
}

const netv = new NetV(configs)
const data = netv.Utils.transformGraphPosition(netv.loadDataset('patents'), 500, 400, 300)
const colorMap = {
  patent: { r: 102, g: 194, b: 165, a: 1 },
  inventor: { r: 252, g: 141, b: 98, a: 1 },
  assignee: { r: 141, g: 160, b: 203, a: 1 }
}

const radius = (x) => {
  const transformer = (n, k) => 0.5 * Math.max(3, k * n**0.5)
  switch (x.type) {
    case 'patent':
      return transformer(x.numCitations, 0.15)
    case 'inventor':
      return transformer(x.numPatents, 0.3)
    case 'assignee':
      return transformer(x.numPatents, 0.1)
  }
}

data.nodes.forEach((node) => {
  const { r, g, b, a } = colorMap[node.type]
  node.fill = { r: r / 255, g: g / 255, b: b / 255, a }
  node.r = radius(node)
})

netv.data(data)
netv.draw()

```

