



Nonnegative spatial factorization applied to spatial genomics



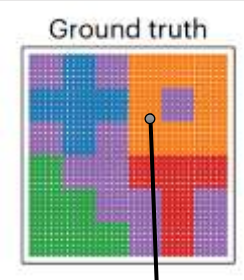
生成空间多组学模拟数据

生成公式-矩阵分解

1	1	1	1	1
		1		
		1		
		1		
1	1	1	1	1

			1	1
1				
	1		1	
1				
			1	

最终数据 = 底噪 + (空间因子 × 空间权重) + (非空间因子 × 非空间权重)



$$\Lambda = \underset{0.2}{bkg} + \underset{(N, L_{sp})}{F} \cdot \underset{(L_{sp}, J)}{W^T} + \underset{(N, L_{ns})}{U} \cdot \underset{(L_{ns}, J)}{V^T}$$

位点	方框	十字
Spot 0	1	0
Spot 1	0	1
Spot 1295	0	1

类型	G1	G2	G3
方框	1	0	1
十字	0	1	0

(2 × 3)

随机指派

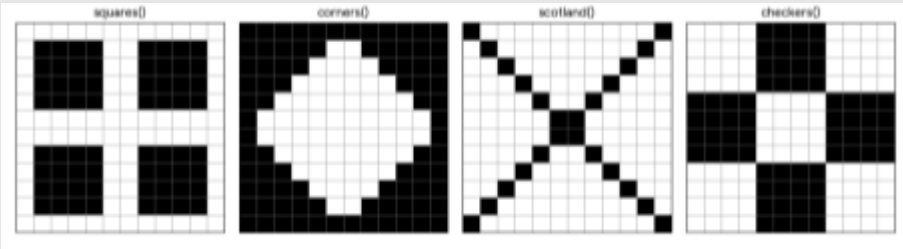
对于矩阵中的每一个元素，程序都会进行一次掷硬币操作：有 20% 的概率生成 1，有 80% 的概率生成 0

nzprob_nsp控制

(i, j)表示第i个空间点的第j个基因是否表达

空间图生成 - quilt() & ggblocks()

12×12 的网格



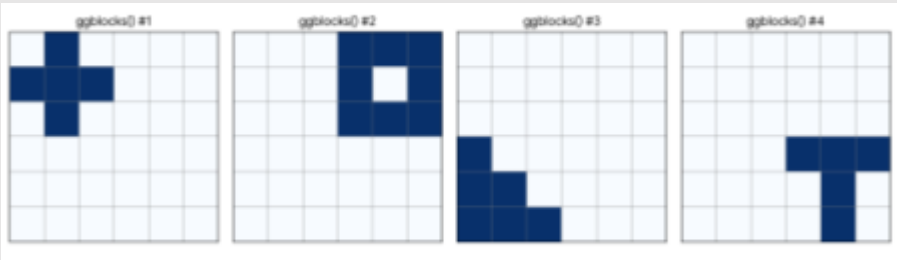
这些图案完全由硬编码写死

```
def corners():
    B = np.zeros([6,6])
    for i in range(6):
        B[i,i] = 1
    A = np.flip(B,axis=1)
    AB = np.hstack((A,B))
    CD = np.flip(AB,axis=0)
    return np.vstack((AB,CD))

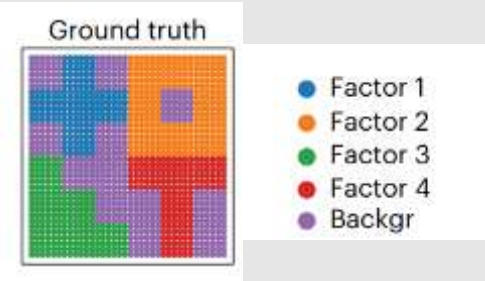
def quilt():
    A = np.zeros([4,144])
    A[0,:] = squares().flatten()
    A[1,:] = corners().flatten()
    A[2,:] = scotland().flatten()
    A[3,:] = checkers().flatten()
    return A #basic block size is 12x12

def ggblocks():
    A = np.zeros([4,36])
    A[0,[1,6,7,8,13]] = 1
    A[1,[3,4,5,9,11,15,16,17]] = 1
    A[2,[18,24,25,30,31,32]] = 1
    A[3,[21,22,23,28,34]] = 1
    return A #basic block size is 6x6
```

6×6 的网格



SpatialGlue



特性	quilt()	ggblocks()
解析难度	解析复杂的重叠纹理和边缘	识别离散、非连续的空间区域
模拟对象	皮层分层、规则排列的组织结构	淋巴滤泡、肿瘤病灶、免疫浸润点
论文用途	展示算法对几何特征的提取能力	展示算法对局部区域聚类的准确性

空间及基因因子生成

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \xrightarrow{\text{克罗内克积}} \begin{bmatrix} 1 \times \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & 0 \times \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \\ 0 \times \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & 2 \times \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

生成空间因子矩阵将上述的小图案，平铺到一个更大的网格上

$$6 \times 6 / 12 \times 12 \rightarrow 36 \times 36 \text{ (1296)}$$

nside可控

$$\begin{matrix} (1296, 1000) \\ (1296, 20) \end{matrix}$$

生成基因表达特征决定了基因的行为: $J = J_{sp} + J_{mix} + J_{ns}$

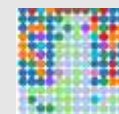
1. J_{sp} 纯空间基因，完美遵循空间图案

$$\Lambda = F \cdot 20 + U \cdot 0$$



2. J_{mix} 混合基因，既受空间图案影响，也受随机噪音影响

$$\Lambda = F \cdot 11 + U \cdot 9$$



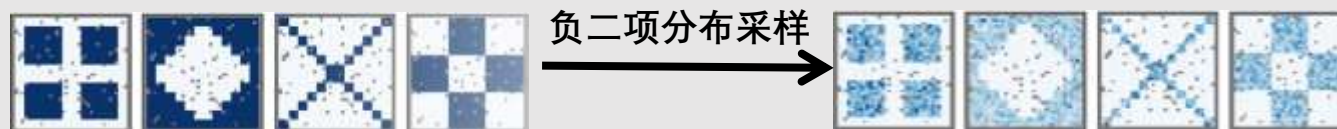
3. J_{ns} 纯噪音基因，完全随机，与位置无关

$$\Lambda = F \cdot 0 + U \cdot 20$$

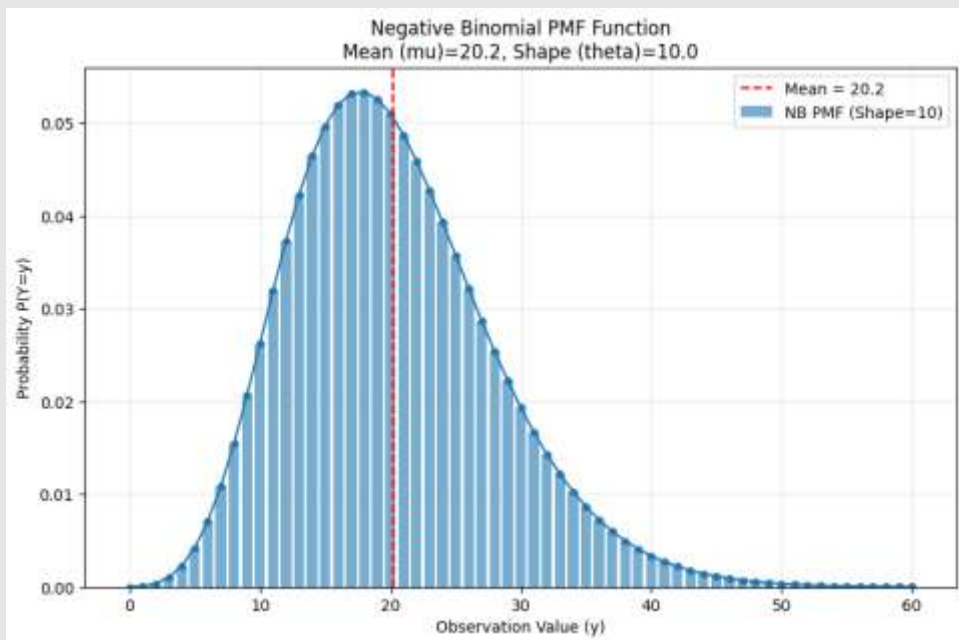


```
def gen_loadings(Lsp, Lns=3, Jsp=0, Jmix=500, Jns=0, expr_mean=20.0,
                 mix_frac_spat=0.55, seed=101, **kwargs):
```

特征融合



合成最终数据把所有东西拼在一起，拿到空间结构 F 和基因权重 W 和 V ，生成随机噪音 U ，计算期望表达量 $Lambda$ ，使用**负二项分布**模拟真实测序数据中常见的过离散现象。最后把坐标、表达量、真实因子打包成一个 *AnnData* 对象。



所以一共得到4种数值：

- 1.信号区 $\lambda = 20.2$ 在这个区域，基因表达很活跃
- 2.背景区 $\lambda = 0.2$ 在这个区域，理论上几乎没有表达
- 3.混合区域 $\lambda = 11.2$ 或 $\lambda = 9.2$ ，这个区域基因表达处于中间态

$$Y_{ij} \sim \text{NegativeBinomial}(\text{mean} = \lambda_{ij}, \text{shape} = 10)$$

$$\text{方差 (Variance)} = \text{均值}(\mu) + \frac{\text{均值}(\mu^2)}{\text{Shape}} \quad \text{其中Shape取10}$$

拓展生成多组学数据

SpaMV将这些点划分为 9 个的方形子区域，每个子区域代表一个独特的空间域

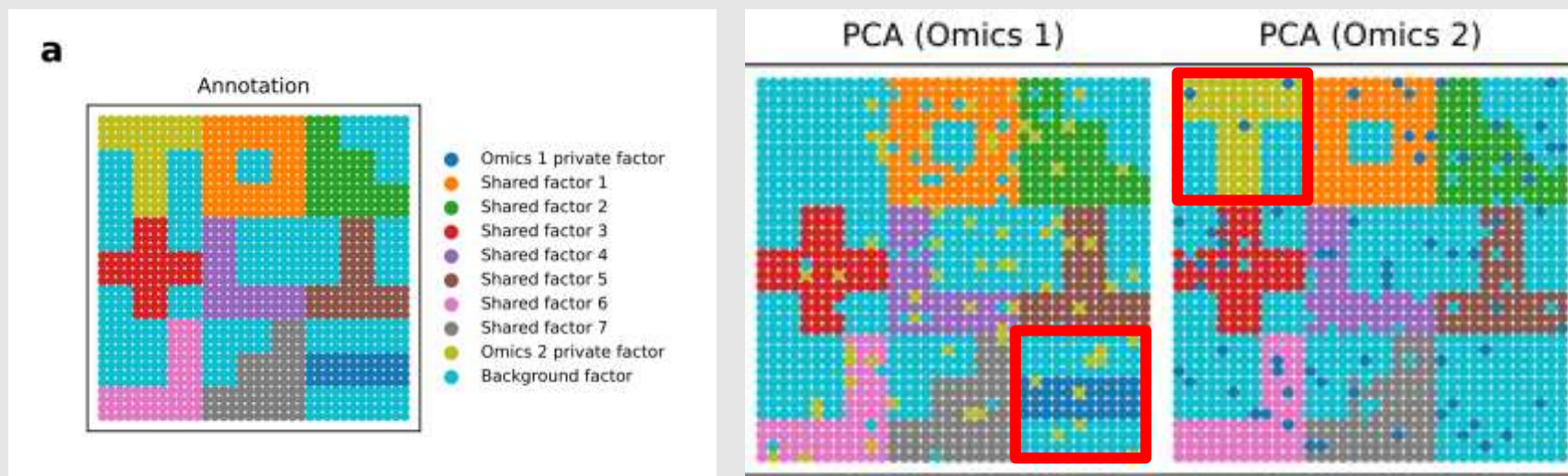
将这些空间域分配给私有因子、共享因子或背景因子

模拟了某些生物学信号是特定组学私有，而另一些是所有组学共享

Omics 1 模拟转录组：生成 1000 个特征，遵循零膨胀负二项分布

Omics 2 模拟其它组学：生成 100 个特征，遵循负二项分布

转录组测序数据通常存在drop-out现象，即很多表达的mRNA没有被捕捉到，导致检测出来的基因表达量为零或者接近零



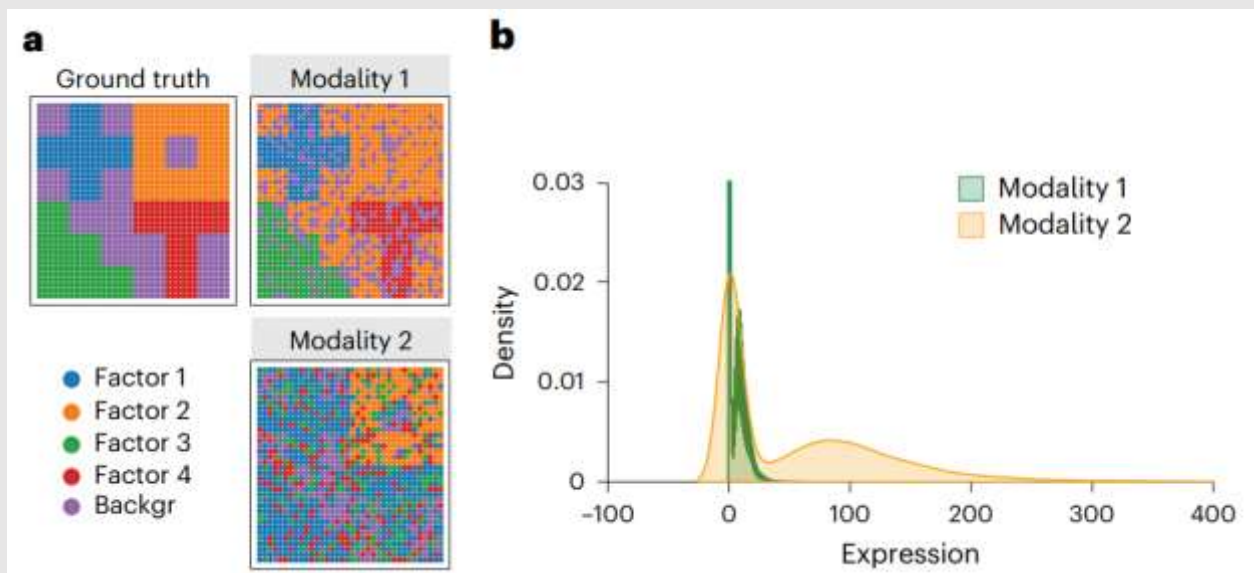
拓展生成多组学数据

SpatialGlue生成的数据包含两类组学，设计了特定的因子来代表解剖结构

因子 1、3、4 由模态 1 决定，而因子 2 仅通过模态 2 识别

模态 1 模拟转录组：遵循零膨胀负二项分布

模态 2 模拟蛋白质组：遵循负二项分布



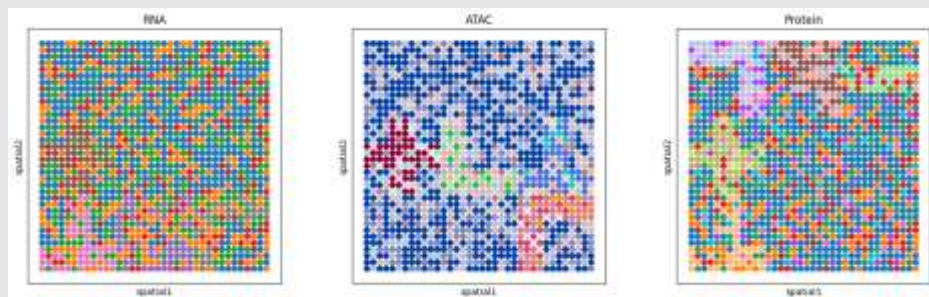
拓展生成多组学数据

可指定组学个数和
每个组学影响的因子

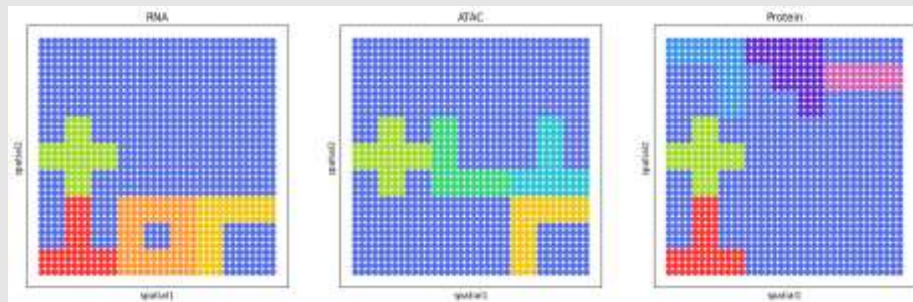
```
{  
  "name": "RNA",  
  "params": {  
    "Jsp": 200, "Jmix": 200, "Jns": 200,  
    "expr_mean": 20.0,  
    "nb_shape": 10.0,  
    "dropout_rate": 0.5  
  },  
  "active_factors": [0, 1]  
},
```

<https://github.com/Fengwenzhe2005/Generate-simulated-spatial-multi-omics-data.git>

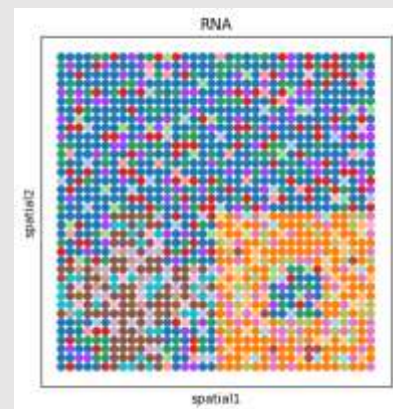
参数差异



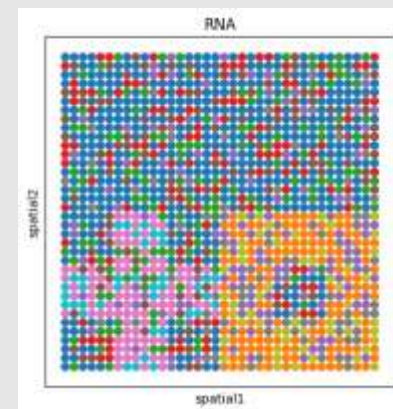
最后输出的leiden聚类结果



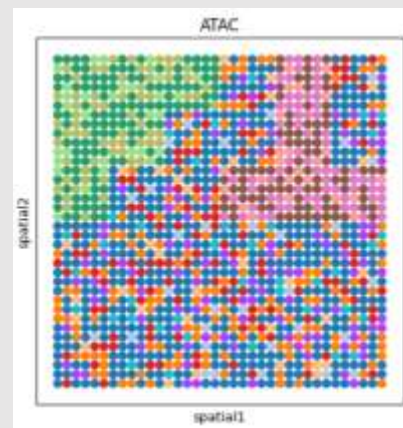
空间因子



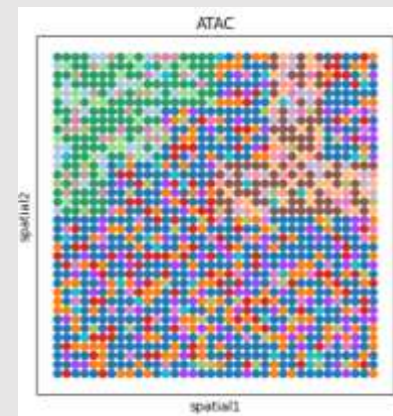
dropout_rate=0.0



dropout_rate=0.5



Jsp=700, Jmix=500, Jns=300



Jsp=500, Jmix=500, Jns=500