基于流形嵌入的宏基因组叠连群分箱方法研究

何 翀[1] 王美丽[1,2,3] 景 旭[1*]

[1](西北农林科技大学信息工程学院 陕西 咸阳 712100)

[2](西北农林科技大学农业农村部农业物联网重点实验室 陕西 咸阳 712100)

[3](西北农林科技大学陕西省农业信息感知与智能服务重点实验室 陕西 咸阳 712100)

摘 要 宏基因组组装往往只能得到较长片段的叠连群,无法恢复完整的基因组。现有的一些分箱方法并未充分挖掘叠连群序列组成和样本覆盖度内部结构信息。开发了基于流形嵌入的宏基因组学叠连群分箱方法,可以挖掘出高维数据中内部的非线性结构特征,从而降低数据的维度,提高计算性能。使用流形嵌入的结果估计出初始分箱数,比使用基于单拷贝基因的分箱数初始化方法更为高效。基于序列组成和样本覆盖度信息,流形嵌入更好地表现出了高维数据嵌入空间的内部结构,为分箱器提供了更有效的特征信息。实验对比了其他方法,结果表明所提方法在 SpeciesMock 数据集上达到了最高的准确率(ACC)、归一化互信息(NMI)和归一化兰德指数(ARI)。

关键词 宏基因组 分装 流形嵌入

中图分类号 TP319 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.03.014

METAGENOMICS CONTIG BINNING BASED ON MANIFOLD EMBEDDING

He Chong[1] Wang Meili[1,2,3] Jing Xu[1*]

[1](*College of Information Engineering, Northwest A & F University, Xianyang 712100, Shaanxi, China*)

[2](*Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Northwest A & F University, Xianyang 712100, Shaanxi, China*)

[3](*Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Northwest A & F University, Xianyang 712100, Shaanxi, China*)

Abstract Metagenomics assembling can only obtain long segments of contigs, and cannot restore the complete genomes. Some existing binning methods do not fully mine the internal structure information of sequence composition and sample coverage of contigs. A metagenomics contig binning method based on manifold embedding is developed, which can mine the internal nonlinear structural features in high-dimensional data, so as to reduce the dimension of data and improve computational performance. It used the results of manifold embedding to estimate the initial bin number, which was more efficient than the bin number initialization method based on single copy genes. Based on the sequence composition and sample coverage information, manifold embedding better showed the internal structure of high-dimensional data embedding space, and provided more effective feature information for binning. Compared with other methods, this method achieves the highest ACC, NMI and Ari on the SpeciesMock data set.

Keywords Metagenomics Binning Manifold embedding

0 引 言

微生物是自然界和人类不可或缺的一类生物[1]。近来研究表明,微生物不仅在自然环境中扮演着重要角色,人体内肠道微生物和多种疾病也存在着某种联系[2],如:肝硬化[3]、肥胖症[4]、糖尿病[5]、肿瘤[6],甚至神经行为类疾病和免疫类疾病[7-8]。科学家们一直

收稿日期:2020-01-08。陕西省重点研发计划项目(2019ZDLNY07-02-01,2019NY-167);农村农业部农业物联网重点实验室项目(2018AIOT-09)。**何翀**,硕士生,主研领域:流形学习,数据挖掘,深度学习,生物信息学。**王美丽**,副教授。**景旭**,副教授。

在研究如何更好地培养和研究微生物,但是因为技术条件的限制,99% 以上的微生物仍然是不可培养的[9]。这导致对于大多数的微生物,人们无法使用传统方式研究它们。

高通量测序技术的不断成熟和成本不断的降低,产生了许多不同视角的研究方法。宏基因组学使用高通量测序技术直接对特定环境进行测序分析,不需要对微生物进行培养,极大地扩充了微生物学的研究对象。但是由于测序技术的限制,当前测序得到的读长往往缺失测序对象的来源信息[10],而且宏基因组读长的组装往往也无法得到完整的基因组,只能得到长片段的叠连群,这会对下游分析产生较大的影响。因此,众多研究人员提出宏基因组学叠连群分箱方法以将宏基因组叠连群分发到操作分类单元(Operational Taxonomic Units,OTU)中,为下游分析提供更为明确的研究对象和更方便的操作对象。在叠连群分箱问题中,基于单拷贝基因确定初始化分箱数的效率和分箱器的效率较低。所以,高效准确的叠连群分箱成为了宏基因组学研究中的重要问题之一。

1　相关工作

宏基因组学叠连群分箱方法可分为依赖其他分箱方法的集成方法和不依赖其他分箱方法的独立方法。独立方法是基于某种数学模型来构建分箱器,比如:高斯混合模型(Gaussian Mixture Model,GMM[11])、期望最大化(Expectation Maximization,EM[12])、k-medoid[13]、仿射传播算法(Affinity Propagation,AP)[14]和非负矩阵分解(Non-negative Matrix Factorization,NMF)[15]等。CONCOCT[11]使用叠连群的覆盖度和四核苷酸频率组成来提取聚类特征,接着使用高斯混合模型来对叠连群进行聚类。作为进一步的拓展,CONCOCT[11]可通过叠连群边缘读长的连接信息来进一步地整合前面聚类后的簇。混合高斯模型确实可以聚类得出一些 OTU 单元,然而混合高斯分布这一假设并非在所有数据上都成立。MaxBin[12]、MetaBAT[13]使用叠连群覆盖度和组成信息联合现存微生物的基因组数据学习得到四核苷酸频率概率距离(Tetranucleotide frequency probability distance,TDP) 和覆盖度距离概率(Abundance Distance Probability,ADP),再使用这两者来构建实验数据中叠连群之间的距离,最后使用改进的 k-medoid 方法进行分箱。虽然利用现存的监督数据学习判别距离可以极大地提高分箱器的学习效率,但是监督模型受限于现有的基因组数据,模型学到的四核苷酸频率距离的判别性能也受限(并不能分离出四核苷酸频率相近的分类单元)。BinSanity[14]使用覆盖度信息和仿射传播算法,迭代地将每一条叠连群当作聚类中心,按照叠连群之间的相似度进行分箱。对 AP 算法的聚类结果中高冗余或低完整度的部分,利用四核苷酸频率和 GC 含量等信息进行提纯等后期处理。AP 算法提高了算法的可适应性,减少了用户的输入参数数目,但是由于 AP 算法的时间复杂度很高,可拓展性较差,在实际应用中的可用性较低。文献[16]和文献[17]中使用关联性特征进行分装,但是对于丰度不均匀的数据效果较差。COCACOLA[15]使用非负矩阵分解的方法对叠连群进行分箱。矩阵分解提供了一个不同的分箱视角,使用序列组成、读长的覆盖度、叠连群比对结果和双端读长连接信息来构建特征矩阵,利用 NMF 算法对特征矩阵分解,最后得到每个叠连群的聚类簇。COCACOLA 提出了一个新的可以融合较多信息的宏基因组学分箱框架。BMC3C[18]使用序列组成、覆盖度信息和密码子信息作为叠连群特征。通过多轮 KMeans 聚类得到的结果构建图模型,使用图分割算法进行划分,将划分结果中的子图作为分箱的结果。SolidBin[10]使用半监督学习的方式将约束信息引入谱聚类方法中,根据不同的约束信息拓展出了几种变体,使用归一化切割方法分箱。虽然不同的变体适应于不同的应用场景,然而如何选取恰当的分箱参数 α、β、K 对于用户较为困难。

使用组合策略的集成方法,利用其他软件的聚类结果,对这些结果进行进一步的优化,比如挑选出其中质量或完整度较高的结果,剔除污染度较高的部分,或使用组装前的读长数据对这些分箱好的簇进行读长的募集,以图提高聚类结果的完整度。DASTools[19]使用基于单拷贝基因的评估函数对其他分箱软件的分箱结果进行评分,然后迭代地从中选取最好的结果,最后集成这些结果形成分箱单元。Binning_refiner[20]的思想是将三种方法的分箱结果合并在一起,然后对合并文件进行比对,对比对结果设定一些阈值,筛选出三种方法共有的即提纯后的簇。metaWRAP[21]在分箱上的想法是,比对以上两种方法(DASTools,Binning_refiner),在融合集成其他聚类结果的基础上,对融合成的簇进行读长募集,进一步地提高簇完整度。metaWRAP 扩大了生物学分类单元的范围,因为募集的过程可能会把近同源的读长募集进簇,另外方法过程中涉及很多比对过程,耗时较长。

对于分箱的初始化,常用的方法有下面几种:变分

贝叶斯自动推断[11](Variational Bayesian estimation of a Gaussian mixture,VBG),使用功能单拷贝标记基因来确定初始聚类数或者使用集成聚类的方法确定最佳聚类数[18]。虽然以上分箱器使用这几种方法,但是这些方法存在着一些问题,如 CONCOCT[11] 中使用的变分贝叶斯方法并不总能够得到理想的结果,基于单拷贝标记基因的方法使用了外部程序对原始的叠连群集进行扫描、翻译和识别,耗时较长,同样地,使用多轮聚类方法进行初始化的方法也存在这样的问题。

针对以上分箱器初始化聚类数效率低下,且未充分利用序列特征流形结构的问题,本文提出基于流形嵌入的分箱方法。流形嵌入是一种用于高维数据的非线性降维方法,降维后的数据不但可以反映出数据内部的隐式结构,为机器学习提供更有效的学习特征,同时也可以用于高维数据的可视化。本文的流形分箱方法可以自适应并高效地估计出初始化分箱数,同时高效地完成宏基因组学叠连群的分箱,如图 1 所示。方法描述如下:

(1) 对叠连群数据进行四核苷酸频率和样本覆盖度特征提取,进行归一化后合并得到原数据 X。

(2) 对归一化后的数据进行流形嵌入。学习得到原始数据嵌入的流形 Y。

(3) 对嵌入流形上的数据 Y 进行 VBG 聚类,得到初始化的聚类数 k。

(4) 从 k 开始增量迭代初始化聚类数,以轮廓系数为聚类性能度量指标,得到最佳的聚类数 K。

(5) 以步骤(4)中参数 K 初始化 KMeans 聚类算法并将流形嵌入得到的结果 Y 进行聚类,得到分箱结果。

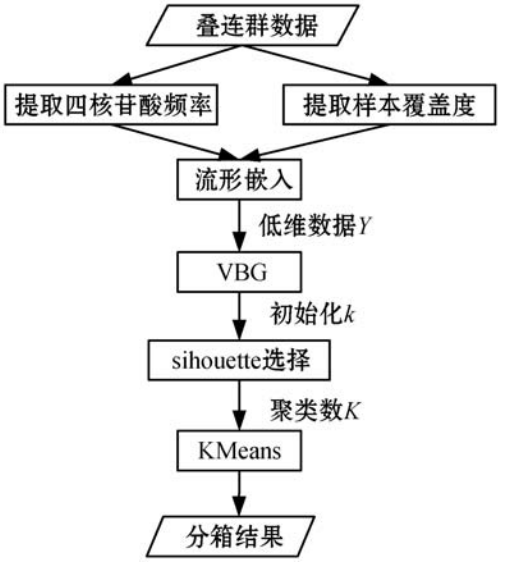


图 1 基于流形嵌入的叠连群分箱方法

2 方 法

2.1 预处理

为了更有效地挖掘叠连群中的信息,同时也为得到每一条叠连群标准的特征描述,需要对叠连群数据进行预处理:提取叠连群的序列组成(即四核苷酸频率)和叠连群的样本覆盖度。

2.1.1 序列组成

K-mer 计数分析在宏基因组学里有许多应用。四核苷酸频率是 K-mer 的一个特化,它统计每一个叠连群中的 4-mer 的计数(回文序列只计一次)。四核苷酸会组成 4^4(256)个词,移除回文词之后,剩余 136 个不同的词。令 $\boldsymbol{P}$ 为四核苷酸计数矩阵,$\boldsymbol{P}\in\mathbf{R}^{n\times k}$,其中:$n$ 为叠连群数量;k 为 136。令 $P_{i,j}$ 为第 i 个叠连群第 j 个词的计数,令 $P'_{i,j}=P_{i,j}+1$,对 $P'_{i,j}$ 进行归一化后得到四核苷酸频率矩阵 $\overline{\boldsymbol{P}}$:

$$\overline{P}_{i,j}=\frac{P'_{i,j}}{\sum_{j=1}^{136}P'_{i,j}} \tag{1}$$

2.1.2 叠连群覆盖度

叠连群丰度的估计是将一个样本中的读长比对到潜在的叠连群集合中。叠连群覆盖度估计即测量每一条叠连群募集读长的数量。叠连群的覆盖度特征(abundance & coverage)期望同一个叠连群在不同时间段内的采样,或者在不同样本中的采样有相似的丰度。令 $\boldsymbol{Q}\in\mathbf{R}^{n\times s}$,其中:$n$ 为叠连群数量;s 为样本数。令 $Q_{i,k}$ 为第 i 个叠连群第 k 个样本的覆盖度,令 $Q'_{i,j}=Q_{i,j}+\frac{100}{L_n}$,$L_n$ 为叠连群的长度。对 $Q'_{i,k}$ 进行样本归一化和叠连群归一化后得到叠连群覆盖度矩阵 $\overline{\boldsymbol{Q}}$:

$$Q''_{i,j}=\frac{Q'_{i,j}}{\sum_{i=1}^{n}Q'_{i,j}} \tag{2}$$

$$\overline{Q}_{i,j}=\frac{Q''_{i,j}}{\sum_{j=1}^{s}Q''_{i,j}} \tag{3}$$

最后将 $\overline{\boldsymbol{P}}$ 和 $\overline{\boldsymbol{Q}}$ 进行拼接$[\overline{\boldsymbol{P}},\overline{\boldsymbol{Q}}]$,得到最后的分箱特征矩阵 $\boldsymbol{X}\in\mathbf{R}^{n\times(136+s)}$,其中:$n$ 为叠连群数量;s 为样本数。

2.2 流形嵌入

均匀流形近似和投影(Uniform Manifold Approximation and Projection,UMAP)[22] 是 2018 年被提出的一种新的流形学习降维方法。定义 $d:\boldsymbol{X}\times\boldsymbol{X}\rightarrow\mathbf{R}_{\geqslant 0}$ 为一种度量,给定一个 k,在 d 下计算 x_i 的近邻集$\{x_{i1},x_{i2},\cdots,x_{ik}\}$,对于每一个 x_i,定义 ρ_i 和 σ_i:

$$\rho_i = \min\{d(x_i, x_{i_j}) \mid 1 \leqslant j \leqslant k, d(x_i, x_{i_j}) > 0\} \quad (4)$$

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k) \quad (5)$$

令 $\overline{G} = (\boldsymbol{X}, E, w)$,其中 $E = \{(x_i, x_{i_j}) \mid 1 \leqslant j \leqslant k, 1 \leqslant i \leqslant N\}$,$w$ 为:

$$w((x_i, x_{i_j})) = \exp\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i} \quad (6)$$

令 $\boldsymbol{A}$ 为 $\overline{G}$ 的加权邻接矩阵:

$$\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}} - \boldsymbol{A} \circ \boldsymbol{A}^{\mathrm{T}} \quad (7)$$

式中:$\circ$ 为 Hadamard 积。可有对应 $\boldsymbol{B}$ 的无权图 G。对 G 的边施加吸引力(*Fattactive*),对边上的顶点施加排斥力(*Frepulsive*),使用模拟退火算法不断减小吸引力和排斥力,最终算法收敛。吸引力和排斥力定义如下:

$$Fattactive = \frac{-2ab \parallel y_i - y_j \parallel_2^{2(b-1)}}{1 + \parallel y_i - y_j \parallel_2^2} w((x_i, x_j))(y_i - y_j) \quad (8)$$

$$Frepulsive = \frac{b(1 - w((x_i, x_j)))(y_i - y_j)}{(\epsilon + \parallel y_i - y_j \parallel_2^2)(1 + \parallel y_i - y_j \parallel_2^2)} \quad (9)$$

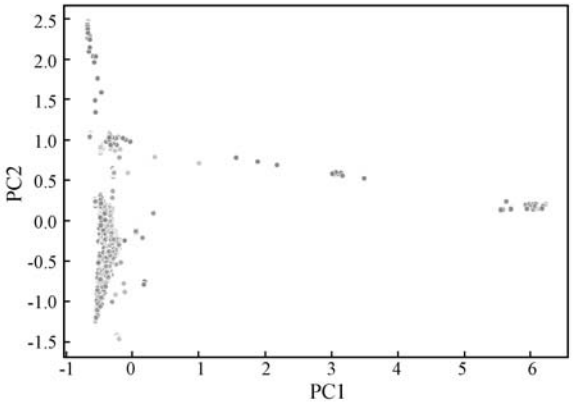
式中:y_i、y_j是低维空间中的坐标;a、b 为参数;ϵ 为无穷小量。

最终得到低维空间下的坐标 $y_i \in \mathbf{R}^d$,所有样本构成的 $Y \in \mathbf{R}^{n \times d}$即流形嵌入得到的结果。

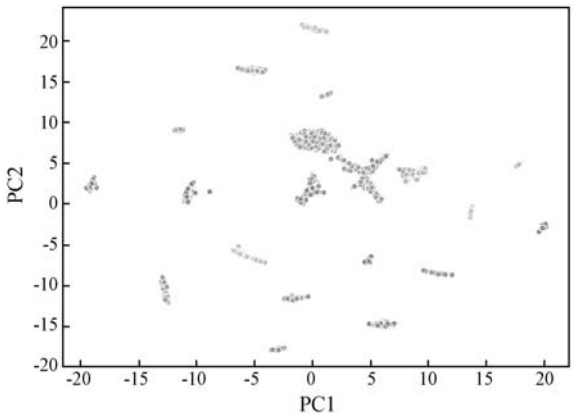
2.3　流形分箱

2.3.1　分箱初始化

从叠连群数据集 StrainMock 的主成分分析降维和流形嵌入降维的可视化结果(如图 2 所示,图中每个点代表一个叠连群)中,发现流形嵌入可以更好地反映出数据集中潜在分箱数。潜在簇的数量是进行聚类数搜索过程良好的初始化,这一点在文献[22]中也有说明。对于流形嵌入得到的结果 Y,使用变分贝叶斯高斯混合模型发现的聚类数 k 作为初始化分箱数。通过一组聚类簇数初始化 KMeans 聚类,并计算对应的 sihouette 系数以评估聚类性能,最终确定最小 sihouette 系数对应的簇数 K 为最终的聚类簇数。



(a) StrainMock 数据集主成分分析



(b) StrainMock 数据集流形嵌入

图 2　StrainMock 数据集可视化

2.3.2　分　箱

对流形嵌入下的结果 Y 进行分箱。为使分箱方法简单、有效并易于解释,以 2.3.1 节得到的最佳初始化分箱数 K 为参数,使用 KMeans 算法对 Y 进行分箱。KMeans 算法将 n 条叠连群划分成 K 个簇,以最小化目标函数:

$$\arg\min \sum_{j=1}^{k} \sum_{y_i \in C_j} \parallel y_i - \mu_j \parallel^2 \quad (10)$$

式中:C_j是第 j 个类簇;$\mu_j = \frac{1}{|y_i|}\sum_{y_i \in C_j} C_j$ 为 C_j;所在簇的中心。

作为拓展,可对不同流形嵌入维度下的结果分别使用 KMeans 算法进行聚类,然后使用集成聚类的一致函数将不同的聚类结果合并,形成最终的分箱结果。

3　实　验

3.1　实验环境和数据集

3.1.1　实验环境

实验运行的环境为 Ubuntu 16.04,48 核 512 GB 内存,Intel(R) Xeon(R) CPU E5-2650 v4,2.20 GHz。本文使用的软件 MaxBin 的版本为 2.2.7,COCACOLA 使用 Python 版本,SolidBin 软件使用 SolidBin-naive 变体。

3.1.2　数据集

(1) SpeciesMock 数据集是基于人类微生物计划(Human Microbiome Project,HMP)中 16S rRNA 数据集构建的,它包括 96 个样本、101 个不同的物种。数据集用来评估在物种水平上的分箱性能。数据集中有 37 628 条叠连群。

(2) StrainMock 数据集是用来测试不同水平上的分箱效果,它包括 64 个样本、20 个不同的物种或菌株。数据集中有 9 417 条叠连群。

3.2 评 估

3.2.1 分箱初始化

对于分箱初始化聚类数,使用了 3.1.2 节中两个标准的分箱数据集进行测试。为了验证流形嵌入对初始聚类数的影响,对每一个数据集分别使用原数据集、PCA 降维结果和 UMAP 嵌入结果进行测试。然后对不同的变换分别使用 XMeans[23]、Meanshift[24] 和 VBG[25] 三种方法计算初始化分箱数,结果越接近于原始的聚类数,说明方法的性能越好。

3.2.2 流形分箱

对流形分箱方法,使用 3.1.2 节中两个标准的分箱数据集进行测试。分别使用 COCACOLA、MaxBin 和 SolidBin 进行分箱,使用标准的评估方法来评估聚类算法的性能:准确度(*ACC*),归一化互信息(*NMI*)和兰德指数(*ARI*)。

准确度:在聚类中,准确度定义是预测标签和真实标签的最佳匹配。

$$ACC = \max \frac{\sum_{i=1}^{n} \text{Ⅱ}\{y_i = m(c_i)\}}{n} \tag{11}$$

式中:n 是样本数;c_i 和 y_i 是第 i 个样本的预测标签和真实标签;Ⅱ为指示函数;m 是匹配函数。

归一化互信息:归一化互信息可以视作对互信息做归一化处理的结果。

$$NMI = \frac{2I(y,c)}{[H(y) + H(c)]} \tag{12}$$

式中:y 是真实标签;c 是聚类的预测标签;H 代表熵;I 是真实标签和预测标签之间的互信息。

归一化兰德系数:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{13}$$

$$RI = \frac{a+b}{C_2^n} \tag{14}$$

式中:a 是在 C、K 中都在同一集合中的样本对数;b 是在 C、K 中都不在同一集合中的样本对数;n 为样本数;RI 为兰德系数;E 为期望。

3.3 实验过程和结果

3.3.1 分箱数初始化

实验首先测试使用贝叶斯高斯混合模型在流形嵌入数据上估计初始化分箱数的效果。实验中使用了原始数据、PCA 变换和 UMAP 变换后的数据集,在数据变换的结果上使用 XMeans、Meanshift 和 VBG 来测试潜在的聚类数即初始化分箱数。

实验结果标注:实验项用"X_Y"来标注,其中下划线前面的字母表示方法,三种方法简写如下:X(XMeans),M(Meanshift),V(VBG)。下划线后面的字母表示数据变换:X(Raw data),P(PCA transform),U(UMAP embedding)。例如,M_X 表示在原数据集上进行 Meanshift 算法的结果。考虑到 UMAP 变换中的随机影响,5 次运行的结果如表 1 和表 2 所示。

表 1 不同方法在 StrainMock 上的分箱数

序号	X_X	X_P	X_U	M_X	M_P	M_U	V_X	V_P	V_U
1	32	32	1	9	9	2	50	50	25
2	32	32	1	9	9	5	50	50	20
3	32	32	1	9	9	6	50	50	19
4	32	32	1	9	9	5	50	50	20
5	32	32	32	9	9	4	50	50	20

表 2 不同方法在 SpeciesMock 上的分箱数

序号	X_X	X_P	X_U	M_X	M_P	M_U	V_X	V_P	V_U
1	32	32	1	16	16	1	200	200	85
2	32	32	1	16	16	1	200	200	93
3	32	32	1	16	16	1	200	200	87
4	32	32	1	16	16	1	200	200	85
5	32	32	1	16	16	1	200	200	88

可以看出,在 StrainMock 数据集中,三种方法在三种变换得到的初始聚类数中最接近真实分箱数(20)的方法就是"V_U",即在流形嵌入结果下使用 VBG 进行聚类。虽然 XMeans 方法较为稳定,但是结果与真实分箱数相差较大,这也说明 XMeans 方法的自适应效果差。Meanshift 方法依赖于带宽参数的设置,对带宽参数极为敏感。在 Meanshift 默认参数下得到的聚类数与真实的聚类数差距较大,在两个数据集下均未得到理想的初始化分箱数。

在两个数据集下使用 VBG 方法进行初始化分箱数估计的实验结果中,StrainMock 数据集下,原始数据集和 PCA 变换后数据集的结果均为 50,这与真实的分箱数相差较大,而在 UMAP 变换下,VBG 得到的分箱数为 19 ~ 25,更接近于真实分箱数。在 SpeciesMock 数据集中结果类似。虽然原始数据集和 PCA 变换后数据集的结果较为稳定,但与真实分箱数(101)相差较大。实验结果中可以发现,流形嵌入对于 VBG 估计出良好的初始化分箱数非常重要,虽然 UMAP 中有随机过程的影响,但是作为初始化分箱数使用,仍比其他方法要好。

3.3.2 流形分箱

如表 3、表 4 所示,在分箱数据中,COCACOLA 在原

始 StrainMock 数据集上表现很好,准确率、归一化互信息和归一化兰德指数分别达到了 0.974 83、0.957 33 和 0.950 78。虽然 COCACOLA 方法较有竞争力,但是本文方法比 COCACOLA 运行所需时间更少,较 MaxBin 方法和 SolidBin 方法相比分箱性能有较大幅度提高,准确率、归一化互信息和归一化兰德指数分别提高 0.095 67、0.027 26 和 0.104 09。

表 3　不同方法在 StrainMock 上的性能

算法	*ACC*	*NMI*	*ARI*
COCACOLA[15]	0.974 83	0.957 33	0.950 78
MaxBin[12]	0.815 12	0.866 73	0.757 05
SolidBin[10]	0.848 57	0.906 28	0.624 36
本文	0.944 25	0.933 54	0.861 15

表 4　不同方法在 SpeciesMock 上的性能

算法	*ACC*	*NMI*	*ARI*
COCACOLA[15]	0.969 49	0.991 94	0.920 62
MaxBin[12]	0.996 47	0.995 34	0.995 33
SolidBin[10]	0.901 38	0.964 03	0.614 27
本文	0.998 64	0.998 13	0.997 23

在 SpeciesMock 数据集中,本文方法分箱性能最好,准确率、归一化互信息和归一化兰德指数分别达到了 0.998 64、0.998 13 和 0.997 23,对比 COCACOLA 的结果有较大程度的性能提高。SolidBin 的性能较差,可能是该方法未充分利用四核苷酸频率和叠连群样本覆盖度信息,过度依赖于其他监督信息的原因。

3.4　结果分析

由表 5 中可以看到,流形嵌入分箱方法占用内存情况。在 StrainMock 数据中的内存占用不是最少的,但较于 SolidBin 来说,本文方法内存占用与 COCACOLA 和 MaxBin 方法同在一个数量级。在 SpeciesMock 数据集,本文方法占用的内存是最少的,且横向对比不同方法在两个数据集上的内存消耗情况,本文方法的增长幅度不大,这说明本文方法具有很好的可拓展性。

表 5　不同方法内存使用　　单位:MB

算法	StrainMock	SpeciesMock
COCACOLA[15]	262	761
MaxBin[12]	884	2 213
SolidBin[10]	4 808	63 129
本文	500	751

不同方法的运行时间如表 6 所示,本文方法与其他分箱方法相比,运行时间最短,在 StrainMock 数据集上用了不到一分钟的时间,在 SpeciesMock 数据集上也仅仅用了 88.21 s。另外三种分箱方法最少运行时间也是本文方法的 4.95 倍。SolidBin 使用的点对点的约束项是造成其内存消耗较大的原因,而 MaxBin 方法中 EM 过程中的迭代是其消耗时间较长的主要原因。从结果中发现,流形嵌入作为叠连群非线性降维方法,为分类器提供了更有效的分箱特征,极大地缩短了分箱器的运行时间,提高了分箱器的计算性能。

表 6　不同方法运行时间　　单位:s

算法	StrainMock	SpeciesMock
COCACOLA[15]	216.73	17 084.59
MaxBin[12]	1 649.25	121 811.57
SolidBin[10]	4 576.99	62 163.44
本文	43.71	88.21

4　结　语

本文将流形嵌入引入到了宏基因组分箱方法中,开发了基于流形嵌入的宏基因组叠连群分箱工具,基于数据流形嵌入,使用变分贝叶斯混合高斯模型可以更高效方便地估算出初始化分箱数,相比较之前使用基于单拷贝基因的方法,所需运行时间较少,运行效率更高,同时具有良好的可拓展性,可运行在个人电脑上。本文提出的基于流形分箱的方法,相比较 MaxBin、COCACOLA、SolidBin,在 SpeciesMock 拥有更高的 ACC、NMI、ARI。本方法尤其适用于菌种水平上、多样本测序的数据。本方法的不足之处是对于样本较少的数据效果较差,即对叠连群的覆盖度信息敏感。在之后的研究里,我们将融入其他先验信息,比如读长连接信息到流形分箱方法中,进一步地提高本文方法的性能。

参　考　文　献

[1] Ewald H S, Ewald P W. Focus: Ecology and evolution: Natural selection, the microbiome, and public health[J]. The Yale Journal of Biology and Medicine, 2018, 91(4): 445.

[2] Russell J A, Dubilier N, Rudgers J A. Nature's microbiome: Introduction[J]. Molecular Ecology, 2014, 23(6): 1225 - 1237.

[3] Tripathi A, Debelius J, Brenner D A, et al. The gut-liver

axis and the intersection with the microbiome[J]. Nature Reviews Gastroenterology & Hepatology,2018,15(7):397 - 411.

[4] Padma M, Vanessa L, Lee M K, et al. The human microbiome and obesity: Moving beyond associations[J]. Cell Host & Microbe, 2017, 22(5):589 - 599.

[5] Johnson E L, Heaver S L, Walters W A, et al. Microbiome and metabolic disease: Revisiting the bacterial phylum Bacteroidetes[J]. Journal of Molecular Medicine,2017,95(1):1 - 8.

[6] Rajagopala S V, Vashee S, Oldfield L M, et al. The human microbiome and cancer[J]. Cancer Prevention Research, 2017,10(4):226 - 234.

[7] Quigley E M M. Microbiota-brain-gut axis and neurodegenerative diseases[J]. Current Neurology and Neuroscience Reports,2017,17(12):94.

[8] Dinan T G, Cryan J F. The microbiome-gut-brain axis in health and disease[J]. Gastroenterology Clinics,2017,46(1):77 - 89.

[9] Torsvik V, Øvreås L. Microbial diversity and function in soil: From genes to ecosystems[J]. Current Opinion in Microbiology,2002,5(3):240 - 245.

[10] Wang Z, Wang Z, Lu Y, et al. Solidbin: Improving metagenome binning with semi-supervised normalized cut[J]. Bioinformatics,2019,35(21):4229 - 4238.

[11] Alneberg J, Bjarnason B S, Bruijn I D, et al. Concoct: Clustering contigs on coverage and composition[EB]. arXiv:1312.4038,2013.

[12] Wu Y W, Simmons B A, Singer S W. Maxbin 2.0: An automated binning algorithm to recover genomes from multiple metagenomic datasets[J]. Bioinformatics,2015,32(4):605 - 607.

[13] Kang D D, Froula J, Egan R, et al. Metabat, an efficient tool for accurately reconstructing single genomes from complex microbial communities[J]. PeerJ,2015,3:e1165.

[14] Graham E D, Heidelberg J F, Tully B J. Binsanity: Unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation[J]. PeerJ,2017,5:e3035.

[15] Lu Y Y, Chen T, Fuhrman J A, et al. Cocacola: Binning metagenomic contigs using sequence composition, read coverage, co-alignment and paired-end read linkage[J]. Bioinformatics, 2017, 33(6):791 - 798.

[16] 张倩倩,曹唱唱,丁啸,等.关联性特征在宏基因组分装中的应用[J].电子器件,2013,36(4):450 - 454.

[17] 丁啸,张倩倩,曹唱唱,等.一种基于关联性特征的宏基因组测序片段分装方法[J].科学通报,2013,58(27):2854 - 2860.

[18] Yu G, Jiang Y, Wang J, et al. Bmc3c: Binning metagenomic contigs using codon usage, sequence composition and read coverage[J]. Bioinformatics, 2018,34(24):4172 - 4179.

[19] Sieber C M K, Probst A J, Sharrar A, et al. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy[J]. Nature Microbiology,2018,3(7):836 - 843.

[20] Song W Z, Thomas T. Binning_refiner: Improving genome bins through the combination of different binning programs [J]. Bioinformatics, 2017, 33(12):1873 - 1875.

[21] Uritskiy G, DiRuggiero J, Taylor J. Metawrap-a flexible pipeline for genome-resolved metagenomic data analysis[J]. Microbiome,2018,6(1):158.

[22] McInnes L, Healy J. UMAP: Uniform manifold approximation and projection for dimension reduction[EB]. arXiv:1802.03426,2018.

[23] Pelleg D, Moore A. X-means: Extending K-means with efficient estimation of the number of clusters[C]//Seventeenth International Conference on Machine Learning,2000:727 - 734.

[24] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2002,24(5):603 - 619.

[25] Blei D M, Jordan M. Variational inference for dirichlet process mixtures[J]. Bayesian Analysis,2006,1(1):121 - 143.

(上接第 13 页)

[17] 蒋明敏.基于 FPGA 的 LCD 伽马校正研究[D].南京:南京林业大学,2016.

[18] He K, Sun J, Tang X. Single image haze removal using dark channel prior[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12):2341 - 2353.

[19] Li J, Lu B L. An adaptive image Euclidean distance[J]. Pattern Recognition, 2009, 42(3):349 - 357.

[20] Liu L L, Yuan Z L, Liu X W, et al. RFID unreliable data filtering by integrating adaptive sliding window and Euclidean distance[J]. Advances in Manufacturing, 2014, 2(2):121 - 129.

[21] 邹承俊.物联网技术在蔬菜温室大棚生产中的应用[J].物联网技术,2013(8):26 - 29,32.

[22] 孙海燕,陈伟国,戴建忠,等.基于物联网技术的智能养蚕温室系统设计[J].蚕桑通报,2017,48(1):51 - 53.

[23] 黎冬媛,朱春媚,莫剑斌.基于 ORM 的农业信息管理系统的设计与实现[J].计算机技术与发展,2011,21(8):204 - 208.