

Pitman-Yor process mixture model for community structure exploration considering latent interaction patterns

Jing Wang(王晶) and Kan Li(李侃)
Citation: Chin. Phys. B, 2021, 30 (**12**): 120518. DOI: 10.1088/1674-1056/ac00a1
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Stationary response of colored noise excited vibro-impact system

Jian-Long Wang(王剑龙), Xiao-Lei Leng(冷小磊), and Xian-Bin Liu(刘先斌)
Chin. Phys. B, 2021, 30 (**6**): 060501. DOI: 10.1088/1674-1056/abf12a

Effects of notch structures on DC and RF performances of AlGaN/GaN high electron mobility transistors

Hao Zou(邹浩), Lin-An Yang(杨林安), Xiao-Hua Ma(马晓华), and Yue Hao(郝跃)
Chin. Phys. B, 2021, 30 (**4**): 040502. DOI: 10.1088/1674-1056/abd470

Stochastic bifurcations of generalized Duffing-van der Pol system with fractional derivative under colored noise

Wei Li(李伟), Mei-Ting Zhang(张美婷), Jun-Feng Zhao(赵俊锋)
Chin. Phys. B, 2017, 26 (**9**): 090501. DOI: 10.1088/1674-1056/26/9/090501

Modeling the capability of penetrating a jammed crowd to eliminate freezing transition

Mohammed Mahmod Shuaib
Chin. Phys. B, 2016, 25 (**5**): 050501. DOI: 10.1088/1674-1056/25/5/050501

Pitman–Yor process mixture model for community structure exploration considering latent interaction patterns*

Jing Wang(王晶) and Kan Li(李侃)†

School of Computer Science, Beijing Institute of Technology, Beijing 100088, *China*

(Received 23 February 2021; revised manuscript received 5 May 2021; accepted manuscript online 13 May 2021)

The statistical model for community detection is a promising research area in network analysis. Most existing statistical models of community detection are designed for networks with a known type of community structure, but in many practical situations, the types of community structures are unknown. To cope with unknown community structures, diverse types should be considered in one model. We propose a model that incorporates the latent interaction pattern, which is regarded as the basis of constructions of diverse community structures by us. The interaction pattern can parameterize various types of community structures in one model. A collapsed Gibbs sampling inference is proposed to estimate the community assignments and other hyper-parameters. With the Pitman–Yor process as a prior, our model can automatically detect the numbers and sizes of communities without a known type of community structure beforehand. Via Bayesian inference, our model can detect some hidden interaction patterns that offer extra information for network analysis. Experiments on networks with diverse community structures demonstrate that our model outperforms four state-of-the-art models.

Keywords: community detection, interaction pattern, Pitman–Yor process, Markov chain Monte–Carlo

PACS: 05.10.Ln, 89.75.Fb, 02.50.Tt **DOI:** 10.1088/1674-1056/ac00a1

1. Introduction

The modern world is highly connected through various interactions that form a large amount of complex networks containing huge valuable data.[1] There are amounts of important applications in the field of complex network, such as the identify influential nodes (Ref. [2]) predicting the evolution of collaboration (Ref. [3]), mitigating the spread of accidents (Ref. [4]) and optimizing the traffic in networks (Ref. [5]). One of the most popular topics of complex networks is the community structure, that is, the vertices divided into groups (also called clusters or blocks) on the basis of a specific criterion or measure.[6]

Newman and Girvan's studies about social and biological networks show that the communities hidden in those datasets are composed of densely connected (or assortative) vertices, while are sparsely connected to those in different communities.[7,8] We call this type of communities with a higher inner connectivity assortative community. Definition of assortative community is ill-suited to the bipartite networks in which the vertices of same type are never connected, e.g., user-item networks, author-paper networks and affiliation networks.[9,10] So in order to group one type of vertices in a bipartite network, we should resort to the methods different from those for the assortative community.[11] For simplicity, we call the communities in bipartite networks bipartite community. Besides, there exists other community structure in which the links are established less frequently inside the communities than those between the different communities, e.g., transcriptional regulatory networks, shareholding networks, and co-occurring word networks.[12,13] We call the communities with higher external connectivity disassortative community. The diverse community structures bring about numerous community detection methods.

However, most existing methods and models focus on detecting certain types of community structures.[14–18] The methods that cannot explore the diverse community structures may fail or lead to biased results, if the network data becomes so complex that we cannot presume a prior for its community structures according to the realistic metadata (e.g., vertices' or interactions' attributes) or the empirical studies (e.g., relevant work).[19] Even though we can presume one type of community structure for a network, it is still possible that other types of community structures are hidden in the network. Therefore what we aim to design is a method that can automatically detect the types of community structures including the number of communities, community sizes and explainable vertex-partitions (see Fig. 1).

One of the most used, flexible and powerful statistical models is the stochastic block models (SBM).[20] It can characterize the assortative community, bipartite community and even disassortative community.[6] There are numerous variants based on SBM or modified SBM, for instance, for the assortative community,[21,22] the bipartite community[10,23] and even the disassortative community.[13] However, in order to achieve

*Project supported by Beijing Natural Science Foundation, China (Grant Nos. L181010 and 4172054), the National Key R&D Program of China (Grant No. 2016YFB0801100), and the National Basic Research Program of China (Grant No. 2013CB329605).

†Corresponding author. E-mail: likan@bit.edu.cn

© 2021 Chinese Physical Society and IOP Publishing Ltd

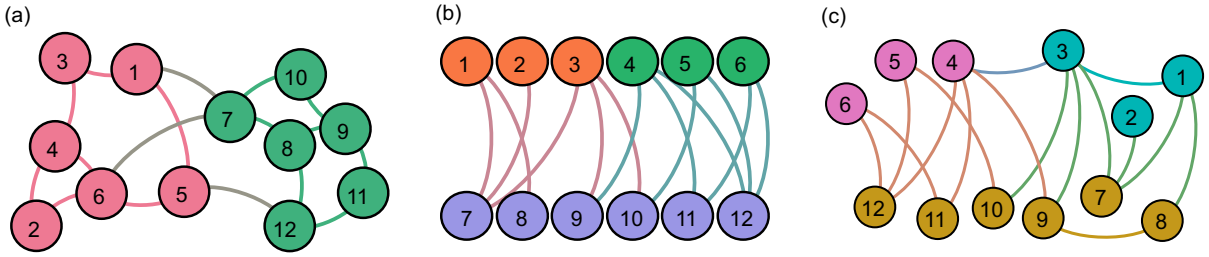


Fig. 1. The toy example of community structure exploration. A competent model should detect three diverse community structures. The colors of vertices represent the different communities found by a community detection method. The colors of links represent the different interaction patterns. (a) Assortative communities; (b) bipartite communities; (c) disassortative communities.

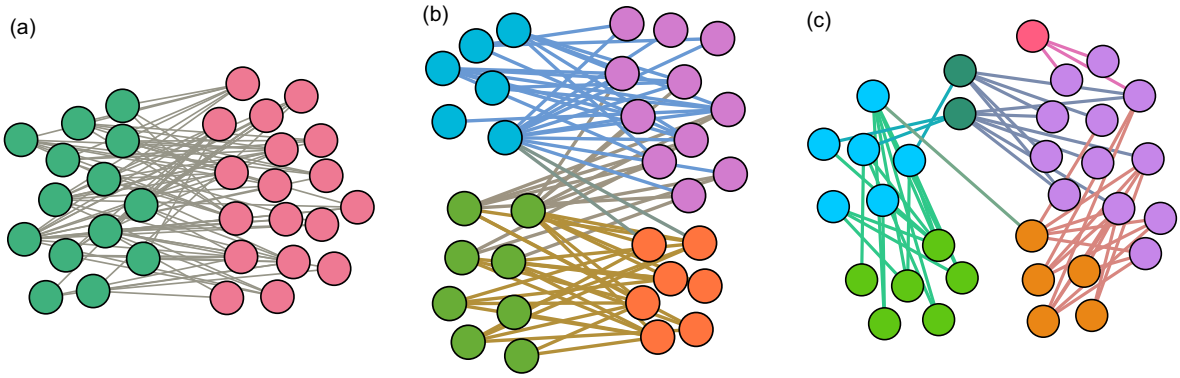


Fig. 2. A toy example of relations between the density of interactions and the vertex partitions: the vertices of the same color own similar interaction patterns. If the interactions are removed, the interaction patterns may alter and so do the vertex partitions. (a) Network of 32 vertices and 89 edges; (b) network of 32 vertices and 71 edges; (c) network of 32 vertices and 53 edges.

a higher likelihood in the parameter inference process, the methods based on the SBM are prone to divide a community even if the vertices in this community have similar behaviors.[24] This character sometimes leads to meaningless and unexplainable results. For instance, a group of people A buy items C, D, and E with high probabilities, while another group of people B buy items C, D, and F with similar probabilities. To prevent overfitting, it is more reasonable to merge these two groups into one community because of similar purchasing habits. This paper will propose an approach to cope with this kind of overfitting.

Community structures can refer to groups of vertices that connect similarly to the rest of the network.[25] Following this idea, we propose an innovative way of defining a community, i.e., incorporating the interaction pattern into community structure, that is to say, all vertices in the same community should share a property in common and the property determines how the vertices interact with the rest of the network. Under the diverse interaction patterns, different types of community structures can be easily represented. Furthermore, the interaction patterns can help our model avoid the kind of overfitting introduced above and lead to explainable results. In this definition, the interactions or links are regarded as statistical units and evidence. The latent interaction patterns will be learned automatically by our model according to the evidence from network data. Specifically, when the interaction patterns are weak, the statistical evidence will be less likely to support grouping the vertices into the same community, and vice versa (for illustration see Fig. 2). The interaction patterns also can be regarded as the features of the communities. But the community exploration through statistical models differs from data classification by feature selection.[26] Our model aims to learn the latent and unknown features described by our assumption, whereas feature selection for data classification aims to learn a subset of features that have already been given.

We then propose a Pitman–Yor process mixture model subtly incorporating interaction pattern in community-structure exploration. Under the rule of the Pitman–Yor process, it can automatically discover reasonable numbers and sizes of communities without any extra information about the community structures and the conversion or projection of the adjacency matrix. Besides, it also can estimate interaction patterns by utilizing Bayesian inference based on the revealed network structure, which offer more information for network analysis. Moreover, experiments have been conducted covering three types of community structures.

Our contributions are summarized as follows:

1. We propose a flexible and effective Pitman–Yor process mixture model inventively incorporating the interaction pattern. It can automatically discover the explainable community structures, even if the interactions are complex.

2. We propose a full-featured collapsed Gibbs sampling inference for community structure exploration. Not only

can it lead to an explainable result but also estimate the latent interaction patterns through Bayesian inference.

3. Detailed experiments with visualizations on 11 real networks of different community structures prove that our model can efficiently reveal complex hidden interaction-patterns and outperforms the state-of-the-art SBM-based models.

2. Related work

The majority of existing approaches are designed for only one type of community structure. For instance, the methods based on random walk for assortative community,[18] the spectral methods for assortative community[16] and for bipartite community,[14] the optimization methods for assortative community,[17] bipartite community,[9] and both of them,[26] the methods based on statistical inference for assortative community[21,22,24,25] and bipartite community,[23] and the deep learning methods for assortative community.[28,29] To adapt one approach for another community structure considerable work is needed. For example, Newman proposed the original modularity to measure assortative community,[30] but it should be modified to be the bipartite modularity for the bipartite community.[9,14] Some other approaches are designed for multiple community structures,[13,27] but they suffer from resolution limit. The approaches based on statistical models employ a principled set of tools for community detection tasks.[6] They can be easily extended to fit the different community structures.[13,23] In this paper, we focus mainly on the statistical models.

One of the most significant statistical models for community detection is the stochastic block model (SBM).[20] Although the SBM has been extensively studied, there still have been numbers of novel approaches based on it in recent years.[22,25,31,32] The block adjacency matrix is a pivotal assumption in the standard SBM. The elements in this matrix define the connection probabilities between two blocks or within a block.[21] To apply statistical inference, each vertex should be assigned to only one block which just corresponds to a community this vertex belongs to. The connection probabilities between two vertices only depend on their blocks, i.e., communities.[21] Different choices of connection probabilities in a block adjacency matrix can lead to different community structures, such as assortative community structure, bipartite community structure, and disassortative community structure.[6]

However, the standard SBM faces two open challenges. Firstly, due to the model's assumption, all vertices within the same community have the same degree distribution, which makes the nodes with the same degree be probably assigned into the same community.[31] The degree-corrected stochastic block model (DCSBM) proposed by Karrer and Newman offers one way to accommodate the degree heterogeneity and allow the vertices of varying degrees to be grouped into one community.[22,33] Because of the broader degree distribution in the communities, DCSBM becomes a widely used generalization of SBM. However, for some networks, DCSBM tends to divide groups even if the group of vertices have similar behaviors. This action can increase the density of links in one block or between two blocks but sometimes may lead to overfitting. Another novel statistical method is the completely random measure stochastic block model (CRMSBM), which incorporates the mass or sociability (similar to degree correction) of vertices and block structure into the completely random measure.[34] However, the number of communities inferred by CRMSBM is closely related to the initial number of communities, which causes CRMSBM unstable performances on some datasets. We will show it in Subsection 4.2.

The second challenge is that the standard SBM requires a specific number of groups K as a prior in order to apply inference. However, K usually varies according to different community structures, network scales and research areas. Using a model selection criterion is one approach to decide which K to choose, e.g., minimum description length (MDL).[35] Regarding the number of groups as a parameter of the model and selecting K directly from the Bayesian inference process is another approach. For instance, use approximate inference, i.e., Markov chain Monte–Carlo (MCMC) to sample community assignments of vertices according to the posterior distribution,[32] or use Bayesian nonparametric models to let the model decide which K (the complexity of the model) to choose, e.g., Chinese restaurant process (CRP) combined with SBM[36,37] and CRP combined with DCSBM.[38] Deep learning for community detection is developing rapidly in recent years. Its powerful feature-representation ability offers us other opportunities to detect community structures by extracting vertices' features.[28] The deep generative latent feature relational model (DGLFRM) proposed by Mehta (Ref. [29]) combines the advantages of variational graph autoencoder and the SBM. It can learn the strength of latent memberships for each vertex and the number of communities can also be learned by the sparse membership representation. However, because of the vague definition of community and unexplainable embedding space, the memberships of the vertices cannot represent the community assignments.

Even though considering the degree correction or the Bayesian nonparametric as a prior or learning the latent membership makes the standard SBM capture various types of community structures,[37] the block that has denser links is just one way to produce communities. As a more general perspective in Abbe's review,[25] community structures can refer

to clusters of vertices that connect similarly to the rest of the networks. In this paper, the interaction pattern we proposed is just corresponding to the above viewpoint and makes our model characterize the three types of community structures or even the mixture of them. In addition, the modeling of degree heterogeneity is also allowed under the assumption of the latent interaction pattern. We will let the inference algorithm choose the numbers and sizes of communities just like these articles (Refs. [37,38]). But we consider the extension of CRP as prior, i.e., the Pitman–Yor process (PYP), which can offer a reasonable community size distribution when the number of communities becomes large.[39]

We next formulate our model specifically and derive an inference algorithm.

3. Model and inference

The notations used in this paper are summarized in Table 1. We use the terms interaction, link and edge interchangeably, and the terms vertex and node interchangeably.

Table 1. Notations frequently used in this paper.

Notation	Description
$\boldsymbol{X}_n$	a network with n interactions or links
X_i	the element of $\boldsymbol{X}_n$, a random variable
d_r	community that vertex r belong to
(r,s)	an interaction or a link between r and s
l	the number of vertices
k	the number of communities

3.1. Model

In this paper, a set of interactions $\boldsymbol{X}_n \triangleq \{X_i\}_i$ constitute the network data, where $i = 1,\ldots,n$. The interaction $X_i = (r,s)$ is represented with two endpoints, vertex r and s.

3.1.1. Interaction pattern

Firstly, as we introduce briefly in Section 1, we consider the variables describing the interaction patterns and how these variables are accommodated to community structures. We assume that the generation of an interaction $X_i = (r,s)$ can be divided into two steps. Firstly, a vertex r starts the head of an interaction X_i, then another vertex s accepts the tail of the interaction X_i. This process can be simply illustrated by a purchasing process in which, for example, a customer selects cereals out of fruits, bread and sandwiches for breakfast. The interaction will be started by the customer as the interaction's head. Next, the prices and the flavors of the cereals will affect the customer's decision. Then a certain cereal product finally "accept" this interaction as its tail. This process can result in the emergence of interaction patterns, say a probability vector $\boldsymbol{\eta}$, with which the customer chooses diverse goods.

Secondly, we assume if the heads are in the same community j, they have the same interaction pattern according to our way of defining the community. For instance, the customers in the community j will choose goods with probability vector $\boldsymbol{\eta}_{j\cdot}$. Specifically, the probability vector $\boldsymbol{\eta}_{j\cdot} = (\eta_{js})_s$ denotes the interaction pattern hidden in the community j, where $j \in (1,\ldots,k)$ denotes a unique community in all k community, and $s \in (1,\ldots,l)$ denotes a unique vertex in the vertex set. The vector $\boldsymbol{\eta}_{j\cdot}$ is normalized, i.e., $\sum_s \eta_{js} = 1$. Different communities will have different $\boldsymbol{\eta}_{j\cdot}$. Just like in the purchasing process, different types of customers will choose the food for breakfast with different interaction patterns $\boldsymbol{\eta}_{j\cdot}$. This definition of interaction pattern $\boldsymbol{\eta}_{j\cdot}$ can characterize diverse community structures including assortative community, bipartite community and disassortative community. We prove that as follows.

Let $\phi_{j_1 j_2}$ denotes the probability of an interaction between two vertices belonging to the community j_1 and j_2 respectively. Actually, $\phi_{j_1 j_2}$ is the same as the elements of the block adjacency matrix in the standard SBM. From the definition of η_{js} we can get

$$\phi_{j_1 j_2} = \sum_{d_s = j_2} \eta_{j_1 s}. \tag{1}$$

Apparently, the link pattern $\boldsymbol{\eta}_{j\cdot}$ can represent all the connection probabilities between two blocks and in a block, which parameterizes SBMs. This interaction-pattern definition is also suitable for both directed and undirected networks. An undirected interaction can be divided into two opposite head-to-tail interactions, which means both endpoints of this interaction will accept the head-to-tail link started by another endpoint.

The parameters discussed above characterize the interaction patterns and their relations with communities. Furthermore, we need a prior for the community-assignment process of vertices. The community-assignment can be considered as a random sequence $\boldsymbol{D}_l \triangleq (D_1,\ldots,D_l)$ that consists of a sequence of positive integers, in which the elements indicate the community a vertex belongs to. This random sequence can also be viewed as a random partition produced by the Pitman–Yor process (PYP).[39] The one-parameter simplified form, i.e., Chinese restaurant process (CRP), has previously been used to determine the numbers and sizes of communities in other SBMs.[36,38] As far as our knowledge, PYP has seldom been employed by the existing community detection methods. Furthermore, the Bayesian nonparametrics including PYP and CRP are flexible and effective priors to represent the community partitions with different numbers and sizes, and also make the inference process easy to implement.[37]

3.1.2. PYP

The PYP is a two-parameter extension to the CRP, which allows heavier-tailed distributions over partitions.[40] PYP can

be interpreted by a CRP metaphor in which the objects are customers in a restaurant, and the groups are the tables at which the customers sit.[41] Imagine a restaurant with infinite tables, each with a unique label. The customers enter the restaurant one after another and then choose an occupied table with probability proportional to the number of occupants minus a discount parameter θ or choose a new vacant table with probability proportional to the concentration parameter α plus $k\times\theta$, where k is the total number of occupied tables. When the discount parameter θ equals to 0, the PYP becomes the CRP. Usually, the PYP and the CRP is used as a prior in the mixture model with an infinite number of latent components, but a finite number of them are used to generate the observed data.[42] In this paper, each community represents a latent component and the interactions are generated according to the interaction patterns associated with the communities of their heads. Compared to the CRP metaphor, the tables denote the communities, a customer denotes the head of an interaction, and the latent components denote the latent interaction patterns. When the $(n+1)$-th customer enters the restaurant with n customers and k occupied tables, the new customer will sit at the table i or a new table with the probability as follows:

$$Pr(d_{n+1}=j|d_1,\ldots,d_n)=\begin{cases}\dfrac{c_i-\theta}{n+\alpha}, & \text{if } j=1,\ldots,k,\\[2ex] \dfrac{\alpha+k\theta}{n+\alpha}, & \text{if } j=k+1,\end{cases} \tag{2}$$

where c_i denotes the number of customers at the table i.

3.1.3. Generative process

The generative process for an interaction set $\boldsymbol{X}_n$, $\boldsymbol{\eta}_{j\cdot}$ and $\boldsymbol{D}_l$ can be constructed as follows.

1. Choose a head r of an interaction uniformly with replacement from a finite vertex set V_l.
2. Assign r to a community d_r according to PYP with the concentration parameter α and discount parameter θ.
3. If the community of r is previously occupied, go to the next step, or if the community is new, draw a probability vector $\boldsymbol{\eta}_{d_r\cdot}$ for this cluster according to a symmetric Dirichlet distribution with parameter $\boldsymbol{\beta}$. In order to generalize our model to fit various interaction patterns, we choose a mild prior, i.e., a symmetric Dirichlet distribution.
4. Then the head r establishes a link to the tail s according to $\boldsymbol{\eta}_{d_r\cdot}$.

The joint probability induced from the above generative process can be written as

$$p(\boldsymbol{X}_n,\boldsymbol{\eta}_{j\cdot},\boldsymbol{D}_l|\alpha,\theta,\boldsymbol{\beta})=p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\eta}_{j\cdot})p(\boldsymbol{\eta}_{j\cdot}|\boldsymbol{\beta})p(\boldsymbol{D}_l|\alpha,\theta). \tag{3}$$

The three factors on the right side can be written as

$$\begin{aligned}p(\boldsymbol{D}_l|\alpha,\theta)&=\frac{\prod_{t=1}^{k-1}(\alpha+t\theta)\prod_{j=1}^{k}\prod_{i=1}^{c_j-1}(j-\theta)}{\prod_{r=1}^{l-1}(\alpha+r)}\\&=\frac{\theta^k\Gamma(\alpha/\theta+k)}{\Gamma(\alpha/\theta)}\prod_{j=1}^{k}\frac{\Gamma(c_j-\theta)}{\Gamma(1-\theta)}\frac{\Gamma(\alpha)}{\Gamma(\alpha+l)},\end{aligned} \tag{4}$$

where c_j denotes the number of vertices in the community j.

$$p(\boldsymbol{\eta}_{j\cdot}|\boldsymbol{\beta})=\prod_{j=1}^{k}\frac{1}{B(\boldsymbol{\beta})}\prod_{s=1}^{l}(\eta_{js})^{\beta_s-1}, \tag{5}$$

where $\boldsymbol{\beta}=(\beta_s,\ldots,\beta_s)$ because of symmetric Dirichlet distribution, $B(\cdot)$ stands for a multivariate Beta function.

$$p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\eta}_{j\cdot})=\prod_{i=1}^{n}\eta_{d_{r_i}s_i}=\prod_{j=1}^{k}\prod_{s=1}^{l}(\eta_{js})^{N_{js}}, \tag{6}$$

where N_{js} denotes the number of tails from block j to s.

Given the observation of $\boldsymbol{X}_n$, the Bayesian inference framework can be used to estimate the unknown parameters. To be brief, we call our model "PYP-IP" (Pitman–Yor process mixture model considering latent interaction pattern).

3.2. Inference and estimation

3.2.1. Parameter and hyperparameter inference

Because the exact inference on two multivariate random variables $\boldsymbol{\eta}_{j\cdot}$ and $\boldsymbol{D}_l$ is intractable, we employ a Markov chain Monte–Carlo (MCMC) algorithm for an approximate estimation. Then the interaction patterns can be estimated through the discovered community structures. k may vary according to different $\boldsymbol{D}_l$ inference on $\boldsymbol{\eta}_{j\cdot}$. In addition, the dimension of $\boldsymbol{\eta}_{j\cdot}$ will grow along with the number of vertices. It results in a rapid expansion of the sample space of $\boldsymbol{\eta}_{j\cdot}$ along with the network size. So tremendous samples are required to reduce the estimation error. With the help of the conjugacy between the Dirichlet and multinomial distributions, we can simplify the joint distribution of the first (Eq. (6)) and the second (Eq. (5)) factor in Eq. (3).

Given $\boldsymbol{D}_l$ and $\boldsymbol{\beta}$ and let $\boldsymbol{\eta}_{j\cdot}\sim\text{Dirichlet}(\boldsymbol{\beta})$, the joint probability of $\boldsymbol{X}_n$ is as follows:

$$\begin{aligned}p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\beta})&=\int p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\eta}_{j\cdot})p(\boldsymbol{\eta}_{j\cdot}|\boldsymbol{\beta})\,\mathrm{d}\boldsymbol{\eta}_{j\cdot}\\&=\int\prod_{j=1}^{k}\frac{1}{B(\boldsymbol{\beta})}\prod_{s=1}^{l}(\eta_{js})^{N_{js}+\beta_s-1}\,\mathrm{d}\boldsymbol{\eta}_{j\cdot}\\&=\prod_{j=1}^{k}\frac{B(\boldsymbol{N}_{j\cdot}+\boldsymbol{\beta})}{B(\boldsymbol{\beta})},\end{aligned} \tag{7}$$

where $\boldsymbol{N}_{j\cdot}$ denotes a vector containing N_{js} in Eq. (6). After integrating out $\boldsymbol{\eta}_{j\cdot}$, we obtain the likelihood

$$p(\boldsymbol{X}_n,\boldsymbol{D}_l|\alpha,\theta,\boldsymbol{\beta})$$

$$= p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\beta})p(\boldsymbol{D}_l|\alpha,\theta)$$
$$= \frac{\theta^k\Gamma(\alpha/\theta+k)}{\Gamma(\alpha/\theta)}\prod_{j=1}^{k}\frac{\Gamma(c_j-\theta)}{\Gamma(1-\theta)}\frac{\Gamma(\alpha)}{\Gamma(\alpha+l)}\frac{B(\boldsymbol{N}_{j\cdot}+\boldsymbol{\beta})}{B(\boldsymbol{\beta})}. \quad (8)$$

Another multivariate random variable $\boldsymbol{D}_l$ represents the latent community assignment. They can be updated by Gibbs sampling scans (sweeps) in light of the CRP metaphor. A vertex label d_r can be updated at each sampling move by fixing the remaining variables in the whole vector $\boldsymbol{D}_l$, where $r\in(1,\ldots,l)$. So

$$\boldsymbol{d}_l^{-i} \triangleq \{d_1,\ldots,d_{i-1},d_{i+1},\ldots,d_l\}$$

denotes the assignments after removing the vertex i. The number of communities k^{-i} can be educed from $\boldsymbol{d}_l^{-i}$. Then the updating rule can be written as a discrete distribution

$$Pr(d_r=j|\boldsymbol{x}_n,\boldsymbol{d}_l^{-r},\alpha,\theta,\boldsymbol{\beta})$$
$$\propto p(\boldsymbol{x}_n|\boldsymbol{d}_l^{-r},d_r=j,\boldsymbol{\beta})\cdot p(d_r=j|\boldsymbol{d}_l^{-r},\alpha,\theta),$$

where $j\in\{1,\ldots,k^{-r}\}$ denotes the label of the j-th community. Let $c_j=\sum_{i=1,i\neq r}^{l}\boldsymbol{I}(d_i=j)$ denote the number of vertices belonging to community j. According to Eq. (7) and the CRP metaphor, the formula above can be written as

$$Pr(d_r=j|\boldsymbol{x}_n,\boldsymbol{d}_l^{-r},\alpha,\theta,\boldsymbol{\beta})$$
$$\propto \frac{\prod_{s=1}^{l}(N_{js}^{-r}+\beta_s)^{\uparrow n_{rs}}}{(\sum\boldsymbol{N}_{j\cdot}^{-r}+\sum\boldsymbol{\beta})^{\uparrow\sum\boldsymbol{n}_{r\cdot}}}$$
$$\times\left(\sum_{i=1}^{k^{-r}}\frac{c_i-\theta}{l-1+\alpha}\delta_i(j)+\frac{\alpha+k^{-r}\theta}{l-1+\alpha}\delta_{k^{-r}+1}(j)\right), \quad (9)$$

where $j\in\{1,\ldots,k^{-r}+1\}$, and $k^{-r}+1$ denotes a new label for a new community. δ_i is Dirac measure, which makes the atom i in the discrete distribution have the weight $\frac{c_j}{l-1+\alpha}$. n_{rs} is the number of interactions from vertex r to vertex s, and usually goes to 1 for unweighted networks. $n^{\uparrow m}=n(n+1)\cdots(n+m-1)$ represents the rising factorial (or ascending factorial function or the Pochhammer symbol), where $n^{\uparrow 0}\equiv 1$ and $n^{\uparrow 1}=n$. Actually, in Eq. (9), we only consider those interactions $(r,\cdot)$ that influence the posterior probability while k varies.

When updating α and θ, we apply the sampling routine via auxiliary variables proposed by Teh.[43] Other methods such as slice sampling can be referred to this article (Ref. [44]). Assuming discount parameter $0\leq\theta\leq 1$ has a prior distribution Beta(a_θ,b_θ) and concentration parameter $\alpha>0$ has a prior Gamma(a_α,b_α), we can deduce three auxiliary variables $Z_1,Z_{2,i}$ and $Z_{3,j}$. Given $\boldsymbol{d}_l$, α and θ, the conditional distribution of the auxiliary variables can be written as follows:

$$Z_1\sim\text{Beta}(\alpha+1,l-1),$$
$$Z_{2,i}\sim\text{Bernoulli}\left(\frac{\alpha}{\alpha+i\theta}\right),$$

where $i\in\{1,\ldots,k\}$, and k denotes the number of blocks induced by $\boldsymbol{d}_l$.

$$Z_{3,j}\sim\text{Bernoulli}\left(\frac{j-1}{j-\theta}\right),$$

where $j\in\{1,\ldots,c_k\}$, and c_k denotes the number of vertices in the community k. Then the posterior distribution conditioned on auxiliary variables can be written as

$$\theta\sim\text{Beta}\left(a_\theta+\sum_{i=1}^{k-1}(1-Z_{2,i}),b_\theta+\sum_{i=1,c_i\geq 2}^{k-1}\sum_{j=1}^{c_i-1}(1-Z_{3,j})\right),$$
$$\alpha\sim\text{Gamma}\left(a_\alpha+\sum_{i=1}^{k-1}Z_{2,i},b_\alpha-\log Z_1\right). \quad (10)$$

For the derivation of the posterior distribution of hyperparameters (formula (10)), it is highly recommended to refer to Teh's article (Ref. [43]).

We can update another hyperparameter β_s according to its fully conditional posterior. Because β_s is independent of α and θ, we can easily deduce the posterior distribution from Eq. (7)

$$p(\boldsymbol{\beta}|\boldsymbol{x}_n,\boldsymbol{d}_l,\alpha,\theta)\propto p(\boldsymbol{x}_n|\boldsymbol{d}_l,\boldsymbol{\beta})p(\boldsymbol{\beta}).$$

By use of Eq. (7), the posterior of β_s has the form

$$p(\beta_s|\boldsymbol{x}_n,\boldsymbol{d}_l,\alpha)$$
$$\propto\prod_{j=1}^{k}\left(\frac{\Gamma(\sum\boldsymbol{\beta})\prod_{s=1}^{l}\Gamma(N_{js}+\beta_s)}{\Gamma(\sum\boldsymbol{N}_{j\cdot}+\sum\boldsymbol{\beta})\prod_{s=1}^{l}\Gamma(\beta_s)}\right)\text{Gamma}(\beta_s|a_\beta,b_\beta). \quad (11)$$

Then slice sampling can be applied to update β_s. One Gibbs scan or sweep of sampling process includes the updating steps defined in Eqs. (9)–(11), respectively.

Let hyperparameters α, β_s have the prior Gamma$(1,1)$ and θ has the prior Beta$(1,1)$ to make the Bayesian inference algorithm automatically estimate these hyperparameters. In addition, the mild priors Gamma$(1,1)$ and Beta$(1,1)$ make the model flexible.

3.2.2. Estimation

Select the best partition. Although after enough iterations our MCMC algorithm can converge, for the analysis task it is still worth finding the best partition according to Bayesian model comparison. Given the observation $\boldsymbol{X}_n$, we can select the best partition by calculating the posterior probability of all the parameters in the model

$$p(\boldsymbol{D}_l,\boldsymbol{\beta},\alpha,\theta|\boldsymbol{X_n})$$
$$\propto p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\beta},\alpha,\theta)p(\boldsymbol{D}_l,\boldsymbol{\beta},\alpha,\theta)$$
$$= p(\boldsymbol{X}_n|\boldsymbol{D}_l,\boldsymbol{\beta})p(\boldsymbol{D}_l|\alpha,\theta)p(\alpha)p(\theta)p(\boldsymbol{\beta}). \quad (12)$$

This posterior probability can be calculated according to Eqs. (4) and (7).

Estimation of interaction pattern After applying the MCMC sampling above, we can obtain the estimated community assignments $\hat{\boldsymbol{d}}_l$ for all the vertices. Simultaneously,

the hyperparameters, i.e., α, θ, and β are estimated, which describe the interaction patterns of the model. Given all these estimated parameters and the training data $\boldsymbol{x}_n$, we apply Bayesian inference for the posterior estimation of the interaction pattern $\hat{\eta}_{js}$, where $j \in \{1,\ldots,\hat{k}\}$. The estimation of $\hat{\eta}_{js}$ equals to the posterior predictive distribution of a link $\tilde{x} = (r,s)$, where the vertex r belongs to the community j.

We first give the prediction rule for a new link. Given an unobserved interaction $\tilde{x} = (r,s)$, we can estimate the probability of this link according to

$$\begin{aligned} & Pr(\tilde{x} = (r,s)|\boldsymbol{x}_n, \boldsymbol{d}_l, \alpha, \theta, \boldsymbol{\beta}) \\ = & \frac{Pr(\tilde{x} = (r,s), \boldsymbol{x}_n|\boldsymbol{d}_l, \alpha, \theta, \boldsymbol{\beta})}{Pr(\boldsymbol{x}_n|\boldsymbol{d}_l, \alpha, \theta, \boldsymbol{\beta})} = \frac{Pr(\tilde{x} = (r,s), \boldsymbol{x}_n|\boldsymbol{d}_l, \boldsymbol{\beta})}{Pr(\boldsymbol{x}_n|\boldsymbol{d}_l, \boldsymbol{\beta})} \\ = & \prod_{j=1}^{k} \frac{B(\boldsymbol{N}^*_{j\cdot} + \boldsymbol{\beta})}{B(\boldsymbol{N}_{j\cdot} + \boldsymbol{\beta})} = \frac{N_{d_r s} + \beta_{\mathrm{s}}}{\sum \boldsymbol{N}_{d_r \cdot} + \sum \boldsymbol{\beta}}, \end{aligned} \tag{13}$$

where k and $\boldsymbol{N}^*_{j\cdot}$ denote the statistics educed from the edge set $\{\tilde{x}, \boldsymbol{x}_n\}$, which are described in Eq. (7).

Then the interaction patterns can be estimated according to Bayesian inference as follows:

$$\begin{aligned} \hat{\eta}_{js} &= Pr(\tilde{x} = (r,s)|\boldsymbol{x}_n, \alpha, \theta, \boldsymbol{\beta}) \\ &= \int Pr(\tilde{x} = (r,s)|\boldsymbol{x}_n, \boldsymbol{d}_l, \alpha, \theta, \boldsymbol{\beta}) p(\boldsymbol{x}_n, \boldsymbol{d}_l|\alpha, \theta, \boldsymbol{\beta}) d\boldsymbol{d}_l \\ &\approx \frac{1}{C} \sum_{t=1}^{T} Pr(\tilde{x} = (r,s)|\boldsymbol{x}_n, \hat{\boldsymbol{d}}_l, \boldsymbol{\beta}) p(\boldsymbol{x}_n, \hat{\boldsymbol{d}}_l|\alpha, \theta, \boldsymbol{\beta}) \\ &= \frac{1}{C} \sum_{t=1}^{T} \frac{N_{js} + \beta_{\mathrm{s}}}{\sum \boldsymbol{N}_{j\cdot} + \sum \boldsymbol{\beta}} p(\boldsymbol{x}_n, \hat{\boldsymbol{d}}_l|\alpha, \theta, \boldsymbol{\beta}), \end{aligned} \tag{14}$$

where $\{\boldsymbol{N}_{j\cdot}\}_j$ denotes the statistics educed from the edge set $\boldsymbol{x}_n$ and $\hat{\boldsymbol{d}}_l$, which are described in Eq. (7), and T denotes the number of iterations after the burn-in period. The normalization coefficient C equals to $\sum_{t=1}^{T} p(\boldsymbol{x}_n, \hat{\boldsymbol{d}}_l|\alpha, \theta, \boldsymbol{\beta})$, which can be calculated via Eq. (8).

3.2.3. Time complexity

The time complexity of our algorithm mainly depends on the MCMC sampler. The runtime per MCMC iteration largely depends on the Gibbs sampling scan for $\boldsymbol{D}_l$ (Eq. (9)). So we can estimate the total runtime as $T = O(\hat{K}LN_{\mathrm{it}})$, where $\hat{K}$ is the estimation of K, L is the number of vertices, and N_{it} is the iteration times. The function of drawing samples from a discrete distribution (described in Eq. (9)) is the essential step of MCMC sampling. The different realization of this function will impact on the runtime notably. So the code optimization for this function can improve the running speed significantly.

4. Experiments and results

4.1. Datasets

Here we apply PYP-IP to the undirected network data of different sizes, types and community structures. Experiments are run on eleven datasets, i.e., characters' co-occurrence network in Les Miserables (Les Mis), the "core" of the email-EuAll network (Eu-core), the protein-protein interaction network in yeasts (Yeast) scientific publication citation network (Cora), the network of hyperlinks between weblogs on US politics (Poli blog), attendance at 14 social events by 18 Southern women (Wom-eve), human musculoskeletal network (Mus-ske), the network denotes which languages are spoken in which countries (Coun-lang), over 100k ratings from MovieLens datasets (ML-100k), adjective-noun word network (Adjnoun), and human transcriptional regulatory network (Tran-reg).

The details of the datasets are shown in Table 2.

Table 2. Dataset details and estimated block numbers.

Network	L	N	K	Structure	$\hat{K}$
Les Mis[45]	77	254	9	Assortative	7.8±0.8
Eu-core[46]	986	24929	42	Assortative	15.6±2.3
Yeast[47]	2284	6646	13	Assortative	59.6±8.7
Cora[48]	2708	5429	7	Assortative	62.6±11.6
Poli blog[49]	1224	19022	2	Assortative	8.6±0.9
Wom-eve[50]	18+14	89	2	Bipartite	4±0
Mus-ske[51]	173+270	953	2	Bipartite	21.4±3.6
Coun-lang[52]	254+614	1255	2	Bipartite	16.2±1.5
ML-100k[53]	610+9724	100836	2	Bipartite	8.2±0.4
Adjnoun[30]	58+54	425	2	Disassortative	2±0
Tran-reg[54]	60+3887	10997	2	Disassortative	20.2±1.1

Self-interactions exist in real network data, such as self-hyperlink and emails to oneself. According to the definition of interaction pattern, a vertex can interact with itself with some probability. So self-loops are allowed in our model. We eliminate self-loops and the isolated vertices for comparison with other models, so that the number of vertices and interactions may be different from the original datasets (include Eu-core, Yeast, Poli blog, ML-100k and Tran-reg). The data pre-processes and the MCMC sampler are implemented by R language and Matlab(R), respectively. All the code are run on a workstation (Intel(R) Xeon(R) CPU E5-2640 v4@2.40 GHz with 64 GB memory).

4.2. Community structure exploration

Community structure exploration can give us some insight into the seemingly disorganized networks. The consideration of interaction pattern makes PYP-IP capable of discovering the latent link patterns.

We apply PYP-IP to group the vertices in the real-world networks without knowing the types of community structures and the number of communities beforehand, to check if PYP-IP can detect the correct community structures. For a fair comparison, all the models in experiments should detect the numbers and sizes of communities automatically. PYP-IP is compared against the infinite relational model (IRM) inferred via Gibbs sampling,[37] the DCSBM

inferred via Metropolis–Hastings sampling with joint merge-split moves,[32] the CRMSBM inferred via Gibbs sampling and Metropolis–Hastings sampling together,[34] and DGLFRM based on variational graph autoencoder combined with SBM.[29] For all the models, the network data will be treated as a square adjacency matrix and all the models do not use any prior information about community structures, such as the number or the sizes of communities. The number of communities K will be initialized by randomly sampling from 1 to $\log(L)$. Given the number of communities K, the community assignments of all vertices will be also initialized randomly. For a fair comparison, DGLFRM will learn the membership of each vertex without extra vertex attributes and each vertex is assigned to the community that has the maximal strength of its community membership. The parameter settings of IRM, DCSBM, CRMSBM and DGLFRM are default as the authors' codes (thanks to these authors generously offering their codes online).

The iteration times of our MCMC sampling are set from 200 to 1000 according to the size of the datasets. All the scores are averaged over 10 restarts.

4.2.1. Assortative community detection

The experiments on the 5 datasets exhibiting assortative community structure will check the models' stability and validity in detecting the densely connected groups. The normalized mutual information (NMI) is used to measure the correlation between clustering results $\boldsymbol{d}'_l$ with the ground truth $\boldsymbol{d}_l$. NMI is calculated as follows:

$$\mathrm{NMI}(\boldsymbol{d}_l, \boldsymbol{d}'_l) = \frac{2\mathrm{MI}(\boldsymbol{d}_l, \boldsymbol{d}'_l)}{H(\boldsymbol{d}_l) + H(\boldsymbol{d}'_l)},$$

where $\mathrm{MI}(\boldsymbol{d}_l, \boldsymbol{d}'_l)$ is mutual information and $H(\boldsymbol{d}_l)$ is entropy of $\boldsymbol{d}_l$. The results are shown in Table 3 with their means and standard variances.

Table 3. NMI score of the clustering results multiplied by 100 with their standard variances. All scores are averaged over 10 restarts.

Model	Les Mis	Eu-core	Yeast	Cora	Poli blog
IRM	61.39±1.78	52.61±0.91	10.98±1.77	25.67±3.06	29.06±0.13
DCSBM	63.69±6.18	65.57±0.90	11.59±0.64	30.97±1.43	31.55±1.02
CRMSBM	51.39±7.30	63.52±1.64	10.05±3.06	11.06±7.22	25.38±14.81
DGLFRM	42.11±5.42	25.62±0.5	6.56±0.24	14.91±0.22	26.96±1.65
PYP-IP	73.26±2.01	65.68±1.41	13.68±2.12	33.66±1.42	43.87±1.09

For assortative community structure, the results show that PYP-IP surpasses the other four models. The performances of all models drop on "Yeast" and "Cora" mainly because of sparser interactions than other networks. DCSBM with joint merge-split moves outperforms IRM and CRMSBM. Both the degree correction and the merge-split moves contribute to the performance of DCSBM. CRMSBM has larger standard variances on "Les Mis", "Cora", and "Poli blog", which shows it is not as stable as the other 4 models. The number of communities discovered by CRMSBM is closely related to the initial number, which causes the unstable performances. Although DGLFRM can learn the latent membership of vertices and the number of communities, sometimes the unexplainable embedding representations are hard to capture community structures.

4.2.2. Bipartite community detection

Bipartite networks have two types of vertices, i.e., the type I and type II, so we assume there are at least two basic communities in bipartite networks. The bipartite community is quite different from the assortative community, that is, there are no interactions within any communities. So models for bipartite community exploration should have two essential functions: 1) distinguish vertices between two types (two basic communities), and 2) group the vertices into the sub-communities. If a model can achieve the first function, the interactions between the bipartite communities should always be detected between two different communities (external links), no matter how the model group the vertices into the sub-communities (see Fig. 1(b)), e.g., woman-event links and person-corporation links should always happen between communities.

Therefore, let the F1 score (F1) help us examine whether the model can achieve the first function. For bipartite community detection, if an interaction is recognized as an external one when the ground truth is external, this sample is true positive (tp), and if an interaction is recognized as an internal one when the ground truth is external, this sample is false negative (fn). F1 score is calculated as follows:

$$\mathrm{F1} = \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})},$$

where "fp" stands for false positive. However, if we exchange the community assignments of a link's head and tail in bipartite networks, the corresponding links may still be external whereas the partition error rate will increase. Here we employ another metric, partition error rate (PER), to measure the partition results for bipartite networks. If a vertex from type I is assigned to type II, PER will increase. It is calculated as follows:

$$\mathrm{PER} = \frac{\#(\text{incorrect assignments})}{l},$$

where "#()" stands for the number of something.

Table 4. The F1 scores and PER of the clustering results multiplied by 100 with their standard variances. All scores are averaged over 10 restarts.

Model	Wom-eve		Mus-ske		Coun-lang		ML-100k	
	F1	PER	F1	PER	F1	PER	F1	PER
IRM	100±0	0±0	92.51±7.83	16.22±7.74	99.97±0.02	2.01±0.37	100±0	0±0
DCSBM	100±0	0±0	97.95±2.3	5.63±4.77	99.90±0.09	6.11±1.44	100±0	0±0
CRMSBM	100±0	0±0	96.48±5.32	9.78±5.97	97.97±3.44	7.35±5.0	99.77±0.05	0.02±0.01
DGLFRM	95.58±2.55	11.25±5.83	95.76±1.99	18.68±4.41	93.81±2.33	19.03±0.16	93.21±2.13	5.90±0.01
PYP-IP	100±0	0±0	99.88±0.14	2.74±0.83	99.86±0.12	2.51±0.19	100±0	0±0

The F1 scores and PERs of bipartite community detection are shown in Table 4 with their means and standard variances. The four datasets of bipartite networks do not have the ground-truth of sub-community structure. According to that paper,[19] we just use the types of nodes as the ground truth. So all the bipartite networks have $K = 2$.

The results show that four statistical models have the ability to automatically detect the bipartite communities on the four datasets. The DCSBM has a higher F1 score and a lower PER on "Coun-lang", whereas PYP-IP has the opposite results. In this situation, we will evaluate the models by PER. Because DCSBM is more likely to exchange the community assignments of vertices in "Coun-lang", which makes some interactions are still external. DGLFRM can detect the external interactions in bipartite networks, but assigns more frequently the two endpoints of interactions to the incorrect vertex type. So its PERs increase a lot.

According to the second function, a model may further split the two basic communities into sub-communities based on the different criteria, e.g., the women attending similar events, i.e., the vertices' behaviors ("Wom-eve" see Fig. 6(b)) or e.g., the collaborative relationship between muscles, i.e., the vertices' functions ("Mus-ske" see Fig. 11(b)). This split by PYP-IP is based on whether the communities have similar interaction patterns and can reveal the latent interaction-pattern information (shown in Subsection 4.4) that is meaningful and explainable.

Although "ML-100k" (99.81% sparsity) is sparser (as normal or square adjacency matrix) than "Mus-ske" (99.03% sparsity), the results (F1 score and PER) are better for "ML-100k" than "Mus-ske", because "ML-100k" contains more interaction-pattern evidence than "Mus-ske" if we regard their network data as the bipartite or rectangular matrices (a user had rated at least 20 movies, but a bone is connected to an average of 5.5 muscles). This result is consistent with our reasonable assumption depicted in Fig. 2.

4.2.3. Disassortative community detection

Because both of the datasets have two types of vertices and this condition is similar to bipartite networks, we can also use the F1 score and partition error rate (PER) to examine the exploration result. In "Adjnoun" dataset, some adjectives can be connected to other adjectives, and some nouns can be connected to other nouns. In "Tran-reg" dataset, some transcription factors not only regulate the target genes but also regulate other transcription factors. So the main difference between disassortative communities and bipartite communities is that the former can have internal interactions, but the later cannot. The F1 scores and PERs of the disassortative community detection are shown in Table 5 with their means and standard variances.

Table 5. The F1 scores and PER of the clustering results multiplied by 100 with their standard variances. All scores are averaged over 10 restarts.

Model	Adjnoun		Tran-reg	
	F1	PER	F1	PER
IRM	79.17±4.61	29.64±10.27	99.65±0.03	0.02±0.01
DCSBM	86.16±1.82	17.86±2.14	99.64±0.02	0.02±0.01
CRMSBM	81.21±3.50	24.64±8.03	99.88±0.01	0.01±0.01
DGLFRM	76.08±4.25	29.82±5.05	94.57±1.29	1.52±0.01
PYP-IP	94.74±1.01	8.57±1.49	99.70±0.01	0.01±0.01

Although both "Adjnoun" and "Tran-reg" have the disassortative community structure, all five models' performances on "Adjnoun" drop more. That is because the percentage of the internal interactions within two communities in "Adjnoun" (38.9%) is significantly larger than in "Tran-reg" (0.86%). However, PYP-IP performs much better than the other four competitors and nearly recovers the disassortative community structure of "Adjnoun" (see Fig. 5(d)).

4.3. Other results

MCMC Convergence. We run our MCMC algorithm on four datasets "Yeast", "Cora" "Adjnoun" and "Mus-ske" to check the convergence of MCMC. Log-likelihoods (Eq. (8)) are recorded as the y-axis and the numbers of iterations are recorded as the x-axis. The results are shown in Fig. 3.

The algorithm can reach the high-probability area of the likelihood for the four datasets with different community structures. When the dataset becomes sparser, e.g., "Mus-ske" 99.03% sparsity and "Cora" 99.85% sparsity, the algorithm needs more iterations to convergence. We will analyze it in the next paragraph.

Hyperparameter β. β is very important to PYP-IP. Its value will go larger when the networks are denser and have more internal interactions. Its value will go smaller when the networks are sparser and have more external interactions. For example, "Adjnoun" is denser and has more internal interactions, so it has the large β_s. The box plot of β_s for the datasets is shown in Fig. 4.

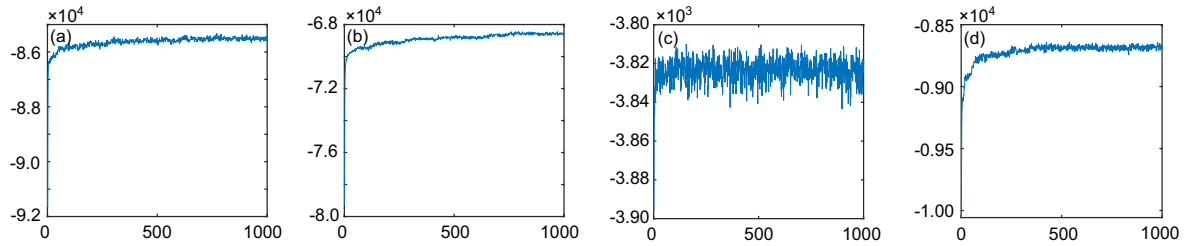


Fig. 3. Log-likelihood after each MCMC iteration. (a) "Yeast"; (b) "Cora"; (c) "Adjnoun"; (d) "Mus-ske".

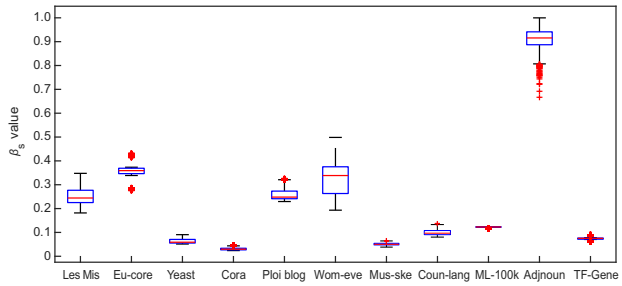


Fig. 4. Box plot for β_s. We keep β_s values of the last 500 MCMC iterations at each of 10 restarts. There are 5000 values in total for each dataset.

Sparsity and community size. PYP-IP inclines to estimate a larger $\hat{K}$ and group the vertices into smaller communities when the networks are sparser (refer to Table 2). Because usually there is not enough evidence to support merging the vertices into a larger community in sparse networks. But if there is more interaction pattern evidence, our model tries to group the vertices that have similar latent interaction-patterns into the same community (see Figs. 5(b) and 6(d)). This character offers reasonable vertex partitions for PYP-IP and it is consistent with our assumption illustrated in Fig. 2. The IRM and DCSBM have similar characters, that is, when the networks are sparser the number of blocks increases so that the density of links in a block or between two blocks increases (refer to Fig. 12). The CRMSBM will infer the number of communities closely related to the initialized number and lead to the unstable performances (see Table 3).

4.4. Visualization

After the community structure exploration, we visualize the clustered networks to check if PYP-IP can explore the hidden interaction-patterns, which will offer extra information for link analysis. The adjacency matrices of networks are showed by rearranging the vertices according to the clustering results. The colored bars on the top and left stand for different communities, which are ordered by community sizes. The vertices in the communities are ordered by their degrees. Parts (a) and (c) of Figs. 5 and 6 show the ground-truth communities, and Part (b) and (d) of these figures show the clustering results by PYP-IP.

Figure 5(b) shows PYP-IP can find a latent link pattern that is a group of vertices having a distinct interaction behavior. These vertices are contacted by other vertices in different communities with higher probabilities than those in other communities. Figure 5(d) shows PYP-IP can detect the interaction patterns that are hidden in "Adjnoun" network, while IRM, DCSBM, and CRMSBM fail to do that. These interaction patterns explored by PYP-IP are not only meaningful but also beneficial to link analysis.

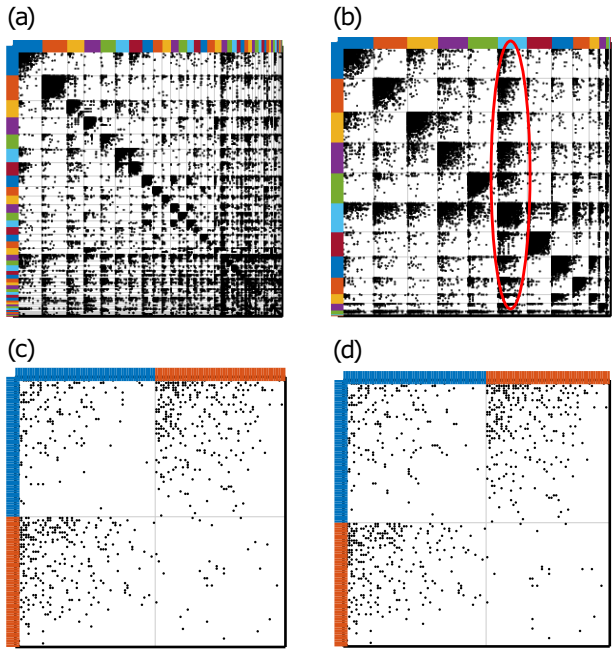


Fig. 5. (a) Ground-truth communities of "Eu-core"; (b) clustering results of "Eu-core" by PYP-IP: in assortative community structure, some distinct interaction-patterns are found. We first estimate η_{js} by Eq. (13) and then calculate $\phi_{j_1 j_2}$ by Eq. (1). The vertices in the red circled community (light blue community in (b)) are connected with a higher average probability ($\bar{\phi}_{\rm l} = 0.16$, where "l" stands for the light blue community) than those in the purple community ($\bar{\phi}_{\rm p} = 0.09$, where "p" stands for the purple community); (c) ground-truth communities of "Adjnoun"; (d) clustering results of "Adjnoun" PYP-IP: comparing (c) and (d), we find the disassortative communities in "Adjnoun" are almost recovered by PYP-IP, while the other comparative models have apparently higher partition error rates.

Figure 6(b) shows the women in orange and yellow communities attend some disjoint events. The two groups of women have the diverse social behaviors. Figure 6(d) shows the blog users' different interaction patterns, that is, more communication with the others outside their community than those in other communities. Although the ground-truth shows the network ("Poli blog") has assortative communities, there actually are "disassortative" communities hidden in assortative community structures. These diverse interaction patterns detected by PYP-IP will offer extra evidence compared to the original partition.

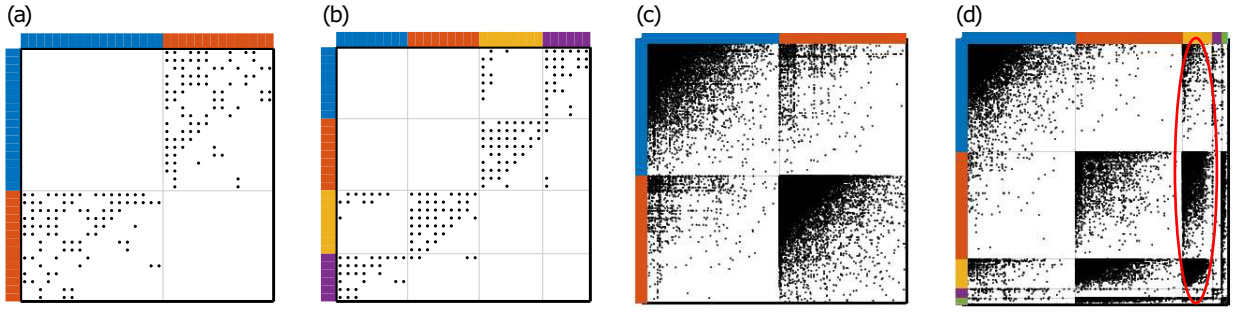


Fig. 6. (a) Ground-truth communities of "Wom-eve"; (b) clustering results of "Wom-eve" by PYP-IP: according to attendance in disjoint events, 18 women can be grouped into 2 communities (the blue and orange) corresponding to two groups of 14 events (the yellow and purple); (c) ground-truth communities of "Poli blog"; (d) clustering results of "Poli blog" by PYP-IP: we first estimate η_{js} by Eq. (13) and then calculate $\phi_{j_1 j_2}$ by Eq. (1). Although the ground truth shows "Poli blog" consists of assortative community structure, there actually is the disassortative community structure (red circle), that is, the users in the orange community interact more often with the yellow community ($\phi_{oy} = 0.47$, where "o" stands for the orange community and "y" stands for the yellow community) than their own community ($\phi_{oo} = 0.32$), and the users in the yellow community interact more often with the orange community ($\phi_{yo} = 0.53$) than their own community ($\phi_{yy} = 0.33$).

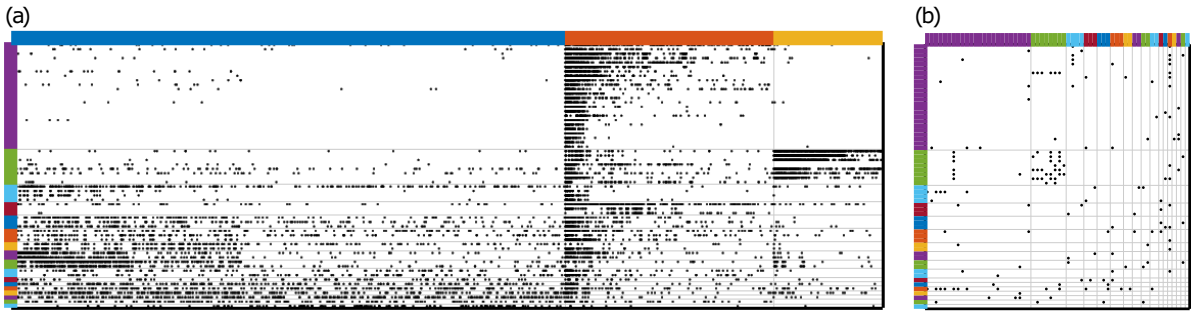


Fig. 7. Clustering results of "Tran-reg": the same color in (a) and (b) represent the same community. In the disassortative community structure, the vertices of the same type can interact with others. From (a) and (b) , we can see in the disassortative community detection PYP-IP finishes two coupled jobs simultaneously: it detects the bipartite communities (a), meanwhile, one type of the vertices in which are grouped just like the assortative community detection (b). (a) Bipartite or rectangular matrix. Columns represent the transcription factors, rows represent the target genes; (b) interactions between the transcription factors.

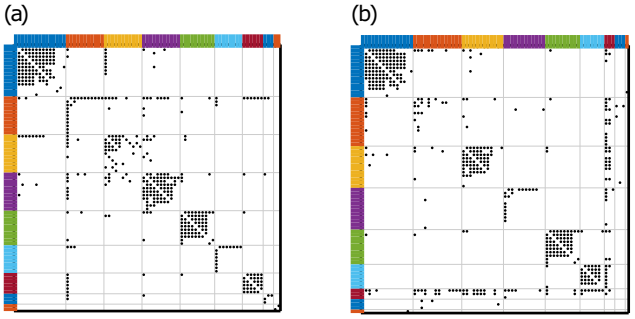


Fig. 8. (a) Ground-truth communities of "Les Mis"; (b) clustering results of "Les Mis" by PYP-IP.

We do not visualize the square adjacency matrix of "Tran-reg", because most of the elements are blank (no interactions between "target genes"). Instead, we visualize the rectangular or bipartite adjacency matrix (x-axis and y-axis in Fig. 7(a) represent "transcription factors" and "target genes" respectively) and the square adjacency matrix of "transcription factors" (Fig. 7(b)).

More results and figures are presented in Figs. 8–11. Parts (a) of Figs. 8–11 show the ground-truth communities, and parts (b) of these figures show the clustering results of our model. These figures show that PYP-IP tries to group the vertices into smaller communities when the networks are sparse and do not have enough interaction-pattern information or evidence. Therefore, the communities are split into smaller ones. When the vertices have similar latent interaction-patterns, PYP-IP groups them into the same community. The following results are also consistent with the assumption proposed in Fig. 2. For some networks, the methods based on SBM such as DCSBM try to divide blocks into smaller ones so that the links in a block or between two blocks become denser (see Fig. 12) and the likelihood becomes higher. However, the vertices that have similar link patterns are separated into different communities and then the link patterns fade away.

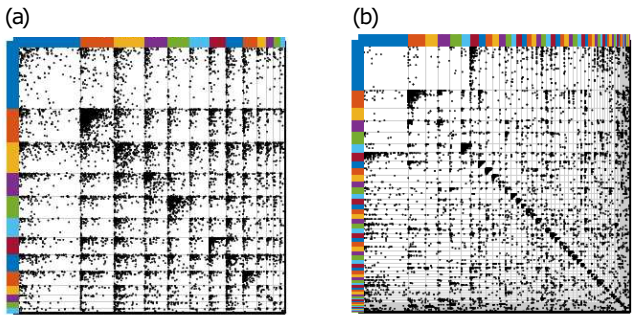


Fig. 9. (a) Ground-truth communities of "Yeast"; (b) clustering results of "Yeast" by PYP-IP.

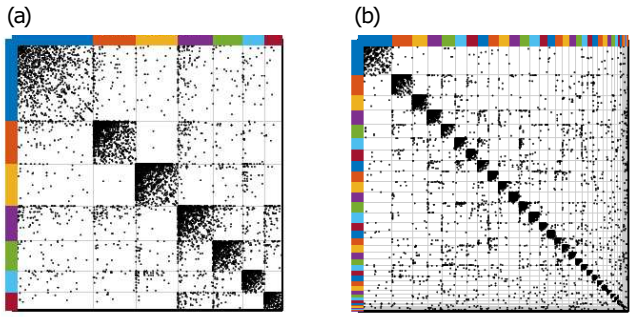


Fig. 10. (a) Ground-truth communities of "Cora"; (b) clustering results of "Cora" by PYP-IP.

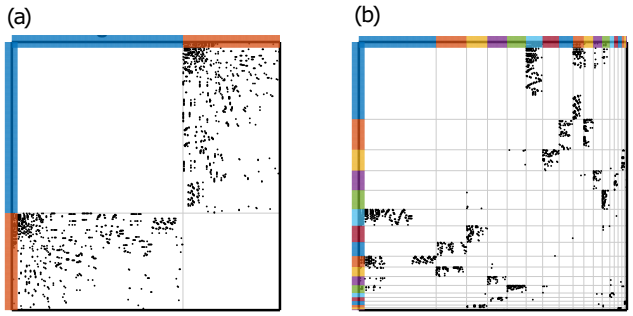


Fig. 11. (a) Ground-truth communities of "Mus-ske"; (b) clustering results of "Mus-ske" by PYP-IP.

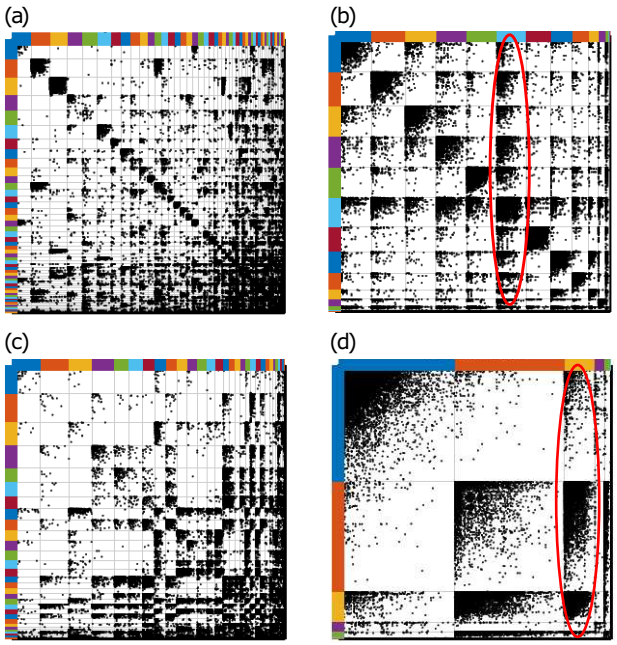


Fig. 12. Blocks detected by DCSBM compared with PYP-IP: the communities detected by DCSBM result in denser links in a block or between blocks, which cause the interaction patterns to fade away. (a) Clustering results of "Eu-core" by DCSBM; (b) clustering results of "Eu-core" by PYP-IP; (c) clustering results of "Poli blog" by DCSBM; (d) clustering results of "Poli blog" by PYP-IP.

5. Conclusions and future work

In this paper, we propose the model, PYP-IP, for detecting communities based on vertices' latent interaction-patterns, and it uses the Bayesian nonparametric, i.e., Pitman–Yor process as a prior for vertex partitions. We prove that PYP-IP can characterize and detect various community structures including assortative community structure, bipartite community structure and disassortative community structure without knowing the type of community structures beforehand. The number and sizes of communities can be estimated automatically through a collapsed Gibbs sampling approach. Then, we evaluate PYP-IP on the networks with different community structures. The experiment results show PYP-IP is competent to explore various community structures. Finally, the visualizations of the adjacency matrix with grouped vertices show some hidden interaction patterns, and can also be revealed by PYP-IP, which will offer extra information for network analysis.

The joint merge-split moves of a group (Ref. [32]) may improve the efficiency of our MCMC algorithm, but the improvement may be a little bit.[55] CRMSBM[34] based on complete random measure owns more theoretical mathematic-background, i.e., random measure, which is an advanced tool that can analyze big data and random events, so it has many merits we should learn. More nonparametric priors can be applied in the model to make it more flexible for sparse networks. The hierarchical extension also can be considered to explore more complex structure in the networks.

References

[1] David E and Jon K 2010 *Networks, Crowds, and Markets: Reasoning About a Highly Connected World* (New York: Cambridge University Press)
[2] Yang Y Z, Hu N and Huang T Y 2020 *Chin. Phys. B* **29** 88903
[3] Hu J W, Gao S, Yan J W, Lou P and Yin Y 2020 *Chin. Phys. B* **29** 88901
[4] Hou G Y, Jin C, Xu Z D, Yu P and Cao Y Y 2019 *Chin. Phys. B* **28** 38901
[5] Pang M B and Huang Y M 2018 *Chin. Phys. B* **27** 118902
[6] Fortunato S and Hric D 2016 *Phys. Rep.* **659** 1
[7] Girvan M and Newman M E J 2002 *Proc. Natl. Acad. Sci. USA* **99** 7821
[8] Newman M E J 2003 *Phys. Rev. E* **67** 026126
[9] Barber M J 2007 *Phys. Rev. E* **76** 66102
[10] Larremore D B, Clauset A and Jacobs A Z 2014 *Phys. Rev. E* **90** 12805
[11] Tackx R, Tarissan F and Guillaume J L 2017 *Complex Networks 2017*, Nov. 2017, Lyon, France p. 278
[12] Newman M E J 2006 *Phys. Rev. E* **74** 36104
[13] Chang C and Tang C 2014 *New J. Phys.* **16** 93001
[14] Feng L, Zhou C and Zhao Q 2019 *Phys. A* **513** 424
[15] Sun H, Ch'ng E, Yong X, Garibaldi J M, See S and Chen D 2018 *Phys. A* **496** 108
[16] Stephan L and Massoulié L 2019 *Proceedings of the 32nd Conference on Learning Theory*, Jun. 25–28, Phoenix, USA p. 2831
[17] Zhang X, Ma Z, Zhang Z, Sun Q and Yan J 2018 *J. Phys. Conf. Ser.* **1069** 12123
[18] Su Y, Wang B and Zhang X 2017 *Sci. Rep.* **7** 41830
[19] Peel L, Larremore D B and Clauset A 2017 *Sci. Adv.* **3** 1602548
[20] Holland P W, Laskey K B and Leinhardt S 1983 *Soc. Networks* **5** 109
[21] Lee C and Wilkinson D J 2019 *Appl. Netw. Sci.* **4** 122
[22] Funke T and Becker T 2019 *PLoS One* **14** 1
[23] Yen T C and Larremore D B 2020 *Phys. Rev. E* **102** 32309
[24] Cherifi H, Palla G, Szymanski B K and Lu X 2019 *Appl. Netw. Sci.* **4** 117
[25] Abbe E 2018 *J. Mach. Learn. Res.* **18** 1
[26] Li Z and Tang J 2021 *Sci. China Inf. Sci.* **64** 192108
[27] Li K and Pang Y 2014 *Neurocomputing* **130** 36
[28] Liu F, Xue S, Wu J, Zhou C, Hu W, Paris C, Nepal S, Yang J and Yu P S 2020 *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Jan. 2021, Yokohama, Japan p. 4981

[29] Mehta N, Duke L C and Rai P *Proceedings of the 36th International Conference on Machine Learning*, June 2019, California, USA pp. 4466–4474
[30] Newman M E J 2004 *Phys. Rev. E* **69** 066133
[31] Gao C, Ma Z, Zhang A Y and Zhou H H 2018 *Ann. Stat.* **46** 2153
[32] Peixoto T P 2014 *Phys. Rev. X* **4** 011047
[33] Karrer B and Newman M E J 2011 *Phys. Rev. E* **83** 016107
[34] Herlau T, Schmidt M N and Mørup M 2016 *Proceedings of the 30th International Conference on Neural Information Processing*, Dec. 2016, Barcelona, Spain, p. 4260
[35] Peixoto T P 2020 *Phys. Rev. E* **102** 12305
[36] Kemp C, Tenenbaum J B, Griffiths T L, Yamada T and Ueda N 2006 *Proceedings of the 21st National Conference on Artificial Intelligence*, July 2006, Boston, USA, p. 381
[37] Schmidt M N and Morup M 2013 *IEEE Signal Process. Mag.* **30** 110
[38] Herlau T, Schmidt M N and Mørup M 2014 *Phys. Rev. E* **90** 32819
[39] Ishwaran H and James L F 2001 *J. Am. Stat. Assoc.* **96** 161
[40] Sato I and Nakagawa H 2010 *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining*, July 2010, New York, USA, p. 673
[41] Griffiths T L and Ghahramani Z 2011 *J. Mach. Learn. Res.* **12** 1185
[42] Gershman S J and Blei D M 2012 *J. Math. Psychol.* **56** 1
[43] Teh Y W 2006 *A Bayesian Interpretation of Interpolated Kneser-Ney* (Singapore)
[44] Johnson M and Goldwater S 2009 *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, May 2009, Boulder, USA, p. 317
[45] Knuth D E 1994 *The Stanford GraphBase: A Platform for Combinatorial Computing* (New York: ACM Press)
[46] Leskovec J and Krevl A 2014 *SNAP Datasets: Stanford Large Network Dataset Collection*, http://snap.stanford.edu/data, [2014]
[47] Bu D, Zhao Y, Cai L, Xue H, Zhu X, Lu H, Zhang J, Sun S, Ling L, Zhang N, Li G and Chen R 2003 *Nucleic Acids Res.* **31** 2443
[48] Lu Q and Getoor L 2003 *Proceedings of the 20th International Conference on International Conference on Machine Learning* Aug. 2003, Washington, USA p. 496
[49] Adamic L A and Glance N 2005 *Proceedings of the 3rd International Workshop on Link Discovery*, Aug. 2005, New York, USA, p. 36
[50] Davis A, Gardner B B and Gardner M R 1941 *Deep South: A Social Anthropological Study of Caste and Class* (Chicago: University of Chicago Press)
[51] Murphy A C, Muldoon S F, Baker D, Lastowka A, Bennett B, Yang M and Bassett D S 2018 *PLOS Biol.* **16** 1
[52] Kunegis J 2013 *Proceedings of the 22nd International Conference on World Wide Web*, May 2013, Rio de Janeiro, Brazil, p. 1343
[53] Harper F M and Konstan J A 2015 *ACM Trans. Interact. Intell. Syst.* **5** 4
[54] Gerstein M B, Kundaje A, Hariharan M, *et al.* 2012 *Nature* **489** 91
[55] Miller K T, Griffiths T L and Jordan M I 2009 *Proceedings of the 22nd International Conference on Neural Information Processing*, Dec. 2009, Vancouver, Canada, p. 1276