BIOS 731
Advanced Statistical Computing
Fall 2020

Lecture 13
Applications of MCMC and SMC

Steve Qin

Review

- Gibbs sampler
- Grouping and collapsing
- Convergence check
- Sequential Monte Carlo
  - Acceptance rejection method
  - Importance sampling

Importance sampling

- *Importance sampling:*
  to evaluate $E_f[h(X)] = \int_{\mathcal{X}} h(x) f(x) dx$
  based on generating a sample $X_1, \cdots, X_n$ from a given distribution g and approximating

$$E_f[h(X)] \approx \frac{1}{m} \sum_{j=1}^{m} \frac{f(X_j)}{g(X_j)} h(X_j)$$

  which is based on

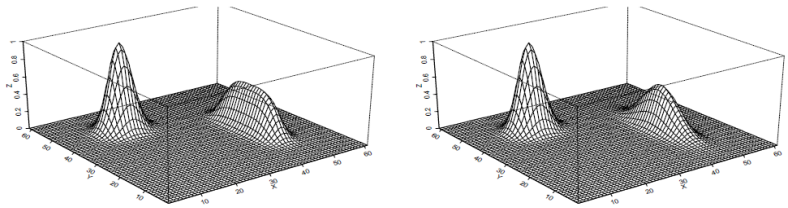
$$E_f[h(X)] = \int_{\mathcal{X}} h(x) \frac{f(x)}{g(x)} g(x) dx$$

3

Another example

$$f(x,y) = 0.5e^{-90(x-0.5)^2-45(y+0.1)^4} + e^{-45(x+0.4)^2-60(y-0.5)^2}$$

(a) (b)



- Both grid-point method and vanilla Monte Carlo methods wasted resources on "boring" desert area.

4

Another example

- Use proposal function

$$g(x,y) \propto 0.5e^{-90(x-0.5)^2-10(y+0.1)^2} + e^{-45(x+0.4)^2-60(y-0.5)^2},$$

with $(x,y) \in [-1,1] \times [-1,1]$, a truncated mixture of bivariate Gaussian

$$0.46\mathcal{N}\left[\begin{pmatrix} 0.5 \\ -0.1 \end{pmatrix}, \begin{pmatrix} \frac{1}{180} & 0 \\ 0 & \frac{1}{20} \end{pmatrix}\right] + 0.54\mathcal{N}\left[\begin{pmatrix} -0.4 \\ 0.5 \end{pmatrix}, \begin{pmatrix} \frac{1}{90} & 0 \\ 0 & \frac{1}{120} \end{pmatrix}\right]$$

Vanilla Monte Carlo

$\hat{\mu} = 0.1307$

$std(\hat{\mu}) = 0.009$

Importance Sampling

$\hat{\mu} = 0.1259$

$std(\hat{\mu}) = 0.0005$

5

Sequential importance sampling

- For high dimensional problem, how to design trial distribution is challenging.
- Suppose the target density of $\mathbf{x} = (x_1, x_2, ..., x_d)$ can be decomposed as

$$\pi(\mathbf{x}) = \pi(x_1)\pi(x_2 \mid x_1)\cdots\pi(x_d \mid x_1, .., x_{d-1})$$

then constructed trial density as

$$g(\mathbf{x}) = g_1(x_1)g_2(x_2 \mid x_1)\cdots g_d(x_d \mid x_1, .., x_{d-1})$$

6

Sequential importance sampling

$$w(\mathbf{x}) = \frac{\pi(x_1)\pi(x_2 \mid x_1)\cdots\pi(x_d \mid x_1,..,x_{d-1})}{g_1(x_1)g_2(x_2 \mid x_1)\cdots g_d(x_d \mid x_1,..,x_{d-1})}$$

Suggest a recursive way of computing and monitoring importance weight. Denote

$$\mathbf{x_t} = (x_1, x_2,..., x_t)$$

then we have

$$w_t(\mathbf{x_t}) = w_{t-1}(\mathbf{x_{t-1}})\frac{\pi(x_t \mid \mathbf{x_{t-1}})}{g_t(x_t \mid \mathbf{x_{t-1}})}$$

7

Sequential importance sampling

- Advantages of the recursion scheme
  - Can stop generating further components of x if the partial weight is too small.
  - Can take advantage of $\pi(x_t \mid \mathbf{x_{t-1}})$ in designing $g_t(x_t \mid \mathbf{x_{t-1}})$
- However, the scheme is impractical since requires the knowledge of marginal distribution $\pi(\boldsymbol{x}_t)$.

8

Sequential importance sampling

- Add another layer of complexity:
- Introduce a sequence of "auxiliary distributions" $\pi_1(x_1)\pi_2(\mathbf{x_2})\pi_d(\mathbf{x})$ such that $\pi_t(\mathbf{x_t})$ is a reasonable approximation of the marginal distribution $\pi(\mathbf{x_t})$, for $t = 1,\ldots,d$ -1 and $\pi_d = \pi$.
- Note the π_d are only required to be known up to a normalizing constant.

9

The SIS procedure

For $t = 2,\ldots,d$,

- Draw $X_t = x_t$ from $g_t(x_t \mid x_{t-1})$, and let $\mathbf{x_t} = (\mathbf{x_{t-1}}, x_t)$
- Compute $u_t = \dfrac{\pi_t(\mathbf{x_t})}{\pi_{t-1}(\mathbf{x_{t-1}})g_t(x_t \mid \mathbf{x_{t-1}})}$ and let $w_t = w_{t-1}\, u_t$
- u_t: incremental weight.
- The key idea is to breaks a difficult task into manageable pieces.
- If w_t is getting too small, reject.

10

An application example of SIS

- Assume
  - Constant population size *N*,
  - Evolve in non-overlapping generation,
  - The chromosomal region is sufficiently small,
  - No recombination,
  - "haplotype": each chromosome only has one parent.

11

Population genetics example

- Notation:
  - E: set of all possible genetic types,
  - μ: mutation rate per chromosome per generation,
  - $P = (P_{\alpha\beta})$: the mutation transition matrix,
  - If a parental segment of type $\alpha \in E$,

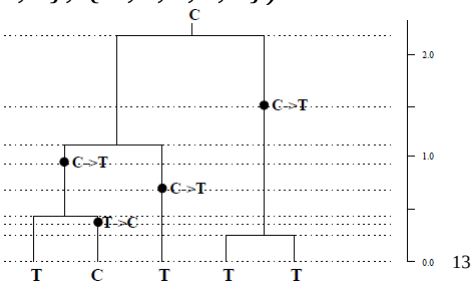
its progeny is $\begin{cases} \alpha & \text{with prob.} 1-\mu, \\ \beta & \text{with prob.} \mu P_{\alpha\beta}. \end{cases}$

12

Example data

- From Stephens and Donnelly (2000)
- E={C,T}
- The history H = (H_{-k}, $H_{-(k-1)}$,…, H_{-1}, H_0)

= ({C}, {C,C}, {C,T}, {C,C,T}, {C,T,T}, {T,T,T}, {T,T,T,T}, {C,T,T,T,T}, {C,T,T,T,T})



13

Coalescence example

- Use H = (H_{-m},…, H_{-1}, H_0) to denote the whole ancestral history (unobserved) of the 5 individuals.
- Compute the likelihood function

$$p_\theta(\boldsymbol{H}) = p_\theta(H_{-k})\,p_\theta(H_{-k+1} \mid H_{-k}) \cdots p_\theta(H_0 \mid H_{-1})\,p_\theta(\text{stop} \mid H_0)$$

$p_\theta(H_{-k}) = \pi_0(H_{-k})$, π_0 is the stationary distribution of P.

$$p_\theta(H_i \mid H_{i-1}) = \begin{cases} \dfrac{n_\alpha}{n}\dfrac{\theta}{n-1+\theta}P_{\alpha\beta} & \text{if } H_i = H_{i-1} - \alpha + \beta \\ \dfrac{n_\alpha}{n}\dfrac{n-1}{n-1+\theta} & \text{if } H_i = H_{i-1} + \alpha \\ 0 & \text{otherwise,} \end{cases}$$

Coalescence calculation

- For i = -(k-1),…,0

$$p_\theta(H_i \mid H_{i-1}) = \begin{cases} \frac{n_\alpha}{n}\frac{\theta}{n-1+\theta}P_{\alpha\beta} & \text{if } H_i = H_{i-1} - \alpha + \beta \\ \frac{n_\alpha}{n}\frac{n-1}{n-1+\theta} & \text{if } H_i = H_{i-1} + \alpha \\ 0 & \text{otherwise,} \end{cases}$$

$$p_\theta(\text{stop}|H_0) = \sum_\alpha \frac{n_\alpha}{n}\frac{n-1}{n-1+\theta}.$$

15

Notations

- n is the sample size at generation $H_{i\text{-}1}$
- n_α is the number of chromosome of type α in the sample.
- θ=2$N\mu$/v.
- N population size.
- v^2 is the variance of the number of progeny of a random chromosome.

16

Strategies to estimate θ

- To get MLE, we need to compute likelihood

$$p_\theta(H_0) = \sum_{\mathcal{H}:\text{compatible with } H_0} p_\theta(\mathcal{H}).$$

- Naïve Monte Carlo won't work because of compatibility issue.
- An alternative is to simulate $\boldsymbol{H}$ backward starting from H_0 and use weight to correct bias.

17

An SIS approach

- Simulate H_{-1}, H_{-1},…, from a trial distribution built up sequentially by revering the forward sampling probability at a fixed θ_0. That is, for i =1,…,k, we have

$$g_t(H_{-t} \mid H_{-t+1}) = \frac{p_{\theta_0}(H_{-t} \mid H_{-t+1})}{\sum_{\text{all } H'_{-t}} p_{\theta_0}(H_{-t} \mid H'_{-t+1})},$$

the final trial distribution

$$g(\boldsymbol{H}) = g_1(H_{-1} \mid H_0) \cdots g_k(H_{-k} \mid H_{-k+1})$$

18

An SIS approach

- By simulating from $g()$ multiple copies of the history, $H^{(j)}$, j=1,…,m, we can approximate the likelihood function as

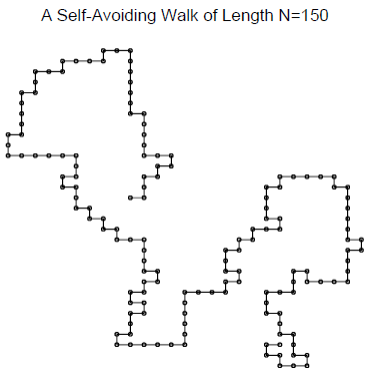
$$\hat{p}_\theta(H_0) = \frac{1}{m}\sum_{j=1}^{m}\frac{p_\theta(H^{(j)})}{g(H^{(j)})}.$$

- Note the choice of θ_0 can influence the final result.

19

Other examples of SIS

- Growing a polymer
  - Self avoid walk
- Sequential imputation for statistical missing data problem.
- More and details of these examples, see Liu 2001.



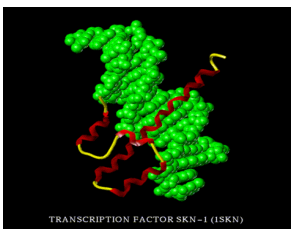
20

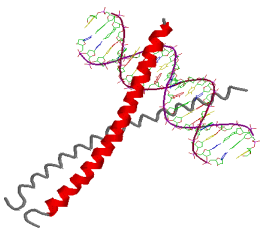
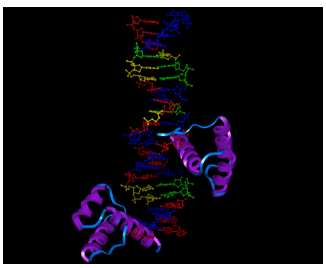
BIOS 731
Advanced Statistical Computing
Fall 2020

Lecture 13
Applications of MCMC and SMC

21

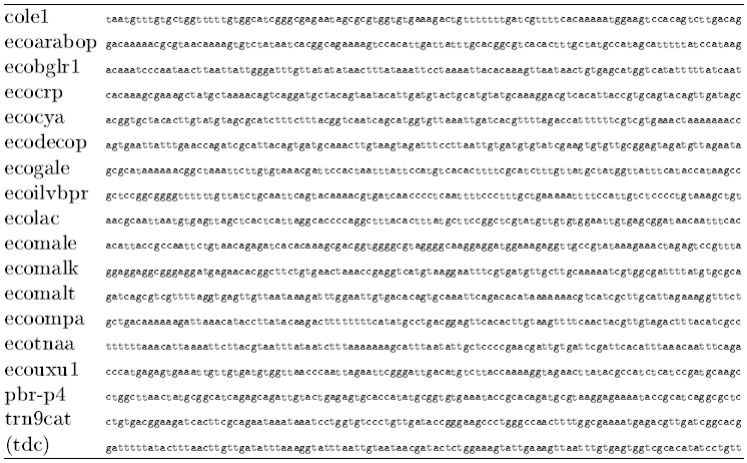
Appliation: Transcription Factor Binding Sites Discovery



22

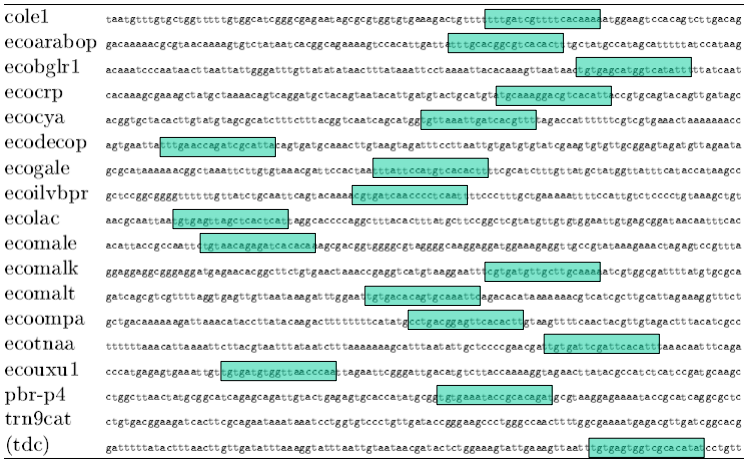
Example: cyclic receptor protein (CRP)



23

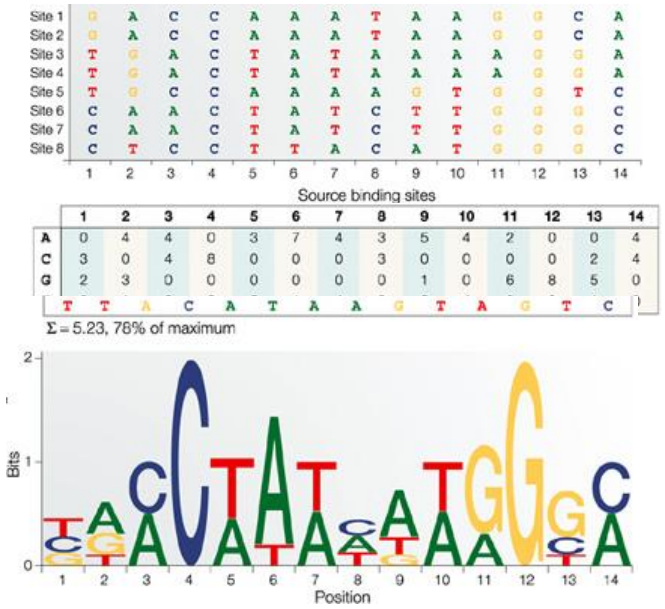
Stormo and Hartzell,

Example: cyclic receptor protein (CRP)



Stormo and Hartzell, 1989

Transcription factor binding site (TFBS)



25

Existing *de novo* motif finding algorithms

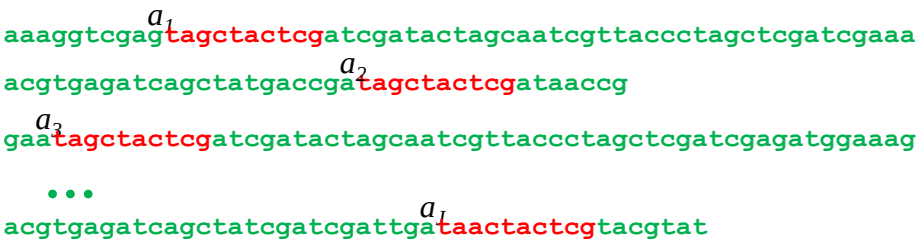
- Consensus — Hertz *et al.* 1990
- Gibbs Motif Sampler — Lawrence *et al.* 1993
- MEME — Bailey and Elkan 1994
- AlignACE — Roth *et al.* 1998
- BioProspector — Liu *et al.* 2001
- MDScan — Liu *et al.* 2002
- Mobydick — Bussemaker *et al.* 2000

…

Review — Tompa *et al.* 2005

26

Motif identification model



Alignment variable $A = \{a_1, a_2 ..., a_J\}$

27

Posterior distributions

- The posterior conditional distribution for alignment variable $\boldsymbol{A}$

$$p(a_j = l \mid \boldsymbol{\theta_0}, \boldsymbol{\Theta}, \boldsymbol{R_j}, \boldsymbol{A}_{-j}) \propto \prod_{k=1}^{4} \theta_{0k}^{h_k(\boldsymbol{R_j})} \prod_{i=1}^{w} \prod_{k=1}^{4} \left(\frac{\theta_{ik}}{\theta_{0k}} \right)^{h_k(r_{j,l+i-1})} \propto \prod_{i=1}^{w} \prod_{k=1}^{4} \left(\frac{\theta_{ik}}{\theta_{0k}} \right)^{h_k(r_{j,l+i-1})}$$

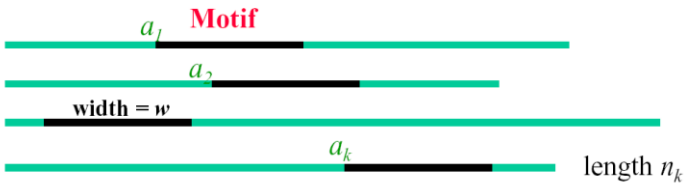
DNA sequence data

$$\boldsymbol{R} = (\boldsymbol{R_1}, ..., \boldsymbol{R_J})$$

Lawrence *et al. Science* 1993, Liu *et al. JASA* 1995

28

Motif Alignment Model



The missing data: Alignment variable: $A=\{a_1, a_2, \ldots, a_k\}$

- Every **non-site positions** follows a common multinomial with $\boldsymbol{p_0}=(p_{0,1},\ldots, p_{0,20})$
- Every position i in the motif element follows probability distribution $\boldsymbol{p_i}=(p_{i,1},\ldots, p_{i,20})$

29

Statistical Model

- Objects:
  - Seq: sequence data to search for motif
  - θ_0: non-motif (genome background) probability
  - θ: motif probability matrix parameter
  - π: site locations
- Problem: $P(\theta, \pi \mid seq, \theta_0)$
- Approach: alternately estimate
  - π by $P(\pi \mid \theta, seq, \theta_0)$
  - θ by $P(\theta \mid \pi, seq, \theta_0)$

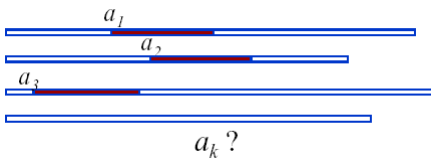
30

The Algorithm

- Initialize by choosing random starting positions
- Iterate the following steps many times;
  - Randomly or systematically choose a sequence to exclude
  - Carry out the predictive-updating step to update the starting position
  - Stop when no more observable changes in likelihood.

31

The Predictive Updating Step



- Compute predictive frequencies of each position *i* in motif

  c_{ij} = count of amino acid type ***j*** at position ***i.***

  c_{0j} = count of amino acid type j in all non-site positions.

  $q_{ij} = (c_{ij}+b_j)/(K\text{-}1+B)$, $B=b_1+ \bullet\bullet\bullet + b_K$ *"pseudo-counts"*

- Sample from the predictive distribution of a_k

$$P(a_k = l+1) \propto \prod_{i=1}^{w} \frac{q_{i,R_k(l+i)}}{q_{0,R_k(l+i)}}$$

32

References

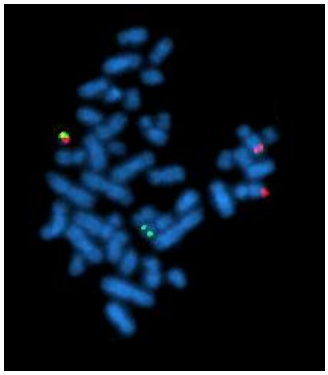
- Lawrence et al. (1993) *Science.*
- Liu, Neuwald and Lawrence (1995) *JASA.*
- Liu and Lawrence (1999) *Bioinformatics.*

33

Infer the 3D shape of chromosomes

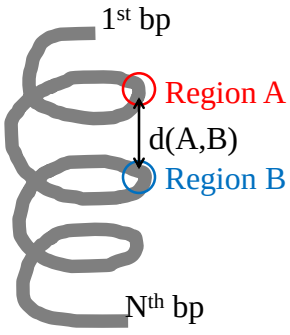
Microscopic Methods

- Fluorescent *in situ* hybridization (FISH)



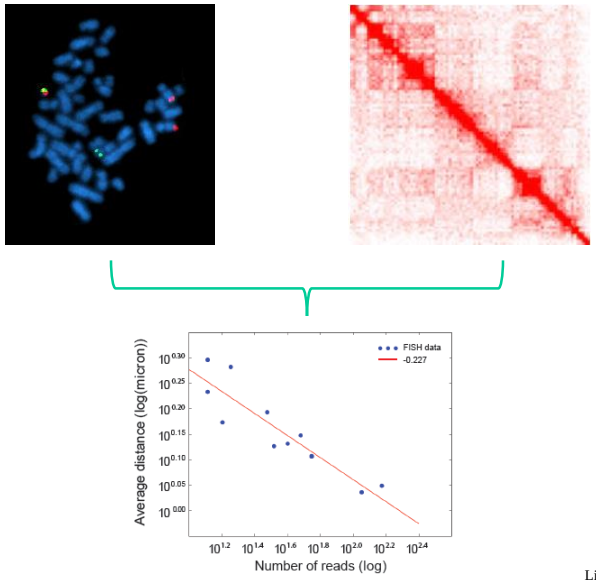
http://en.wikipedia.org/wiki/Cytogenetics

FISH Data Representation



3D chromosomal structure

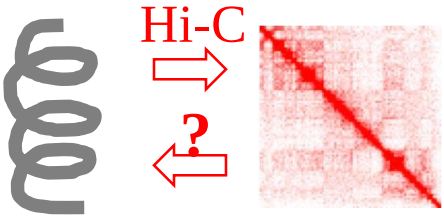
Contact Frequency vs. Spatial Distance



37

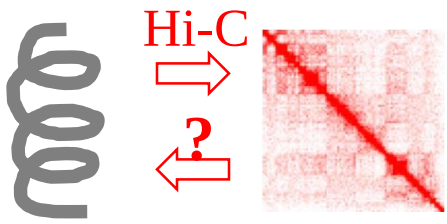
Lieberman-Aiden, et al, 2009

Problem setting



38

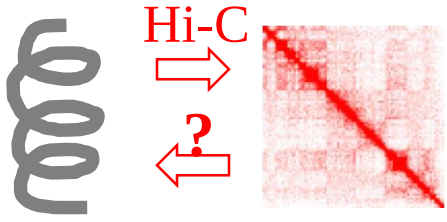
Problem setting



- Challenges:
  - Sequencing uncertainties
  - Biases: enzyme, GC content, mappability

39

Problem setting



- Challenges:
  - Sequencing uncertainties
  - Biases: enzyme, GC content, mappability

40

Yaffe and Tanay, 2011

Beads-on-a-string Representation

ACGTAGCTAGATACTGTAGTGTAGTTTGGAACCTGAGGG

41

Beads-on-a-string Representation

ACGTAGCTAGATACTGTAGTGTAGTTTGGAACCTGAGGG

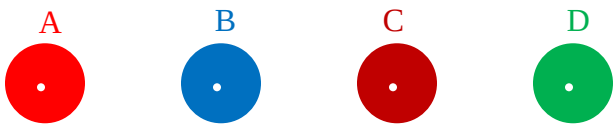
42

Beads-on-a-string Representation

ACGTAGCTAG ATACTGTAGT GTAGTTTGGA ACCTGAGGG

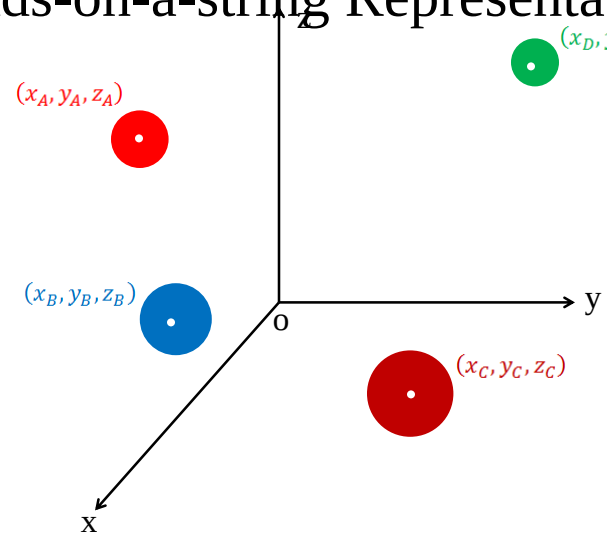
43

Beads-on-a-string Representation



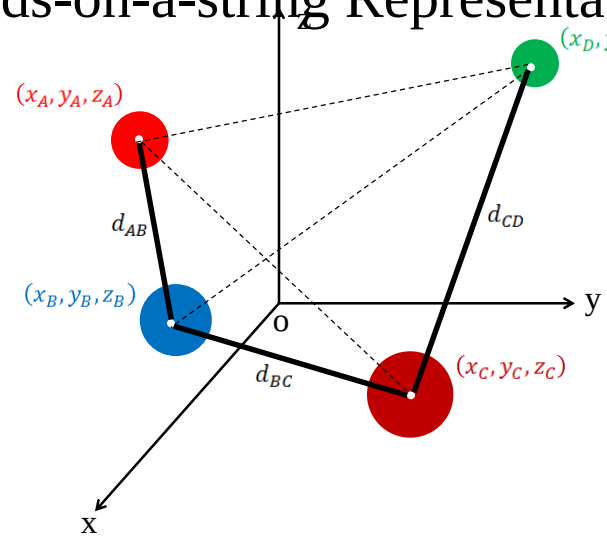
44

Beads-on-a-string Representation



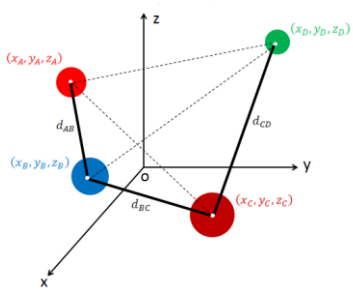
45

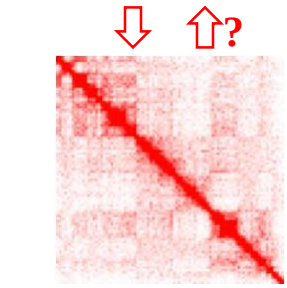
Beads-on-a-string Representation



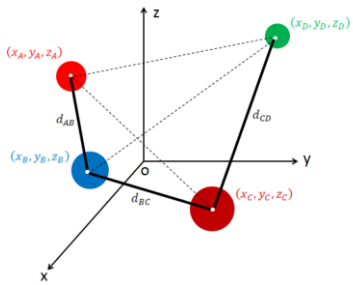
46

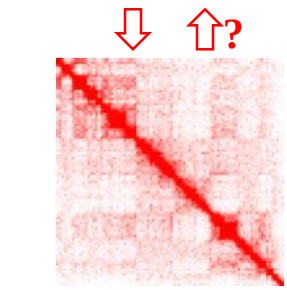
Bayesian Statistical Model





Bayesian Statistical Model



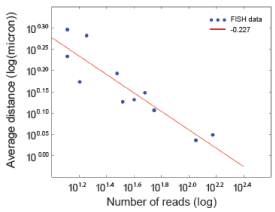


u_{ij} : # of reads between loci i and j
(x_i, y_i, z_i) : Euclidian coordinates of locus i
d_{ij} : spatial distance between loci i and j
e_i : # of enzyme cut site in locus i
g_i : GC content of locus i
m_i : mappability of locus i

Hi-C read counts: population summation

$$u_{ij} \sim Poisson(\theta_{ij})$$

Hi-C read counts vs. spatial distance: log-log linear



$$\log(\theta_{ij}) = \beta_0 + \beta_1 \log(d_{ij}) + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

Lieberman-Aiden, et al, 2009

Bayesian Statistical Model

- Likelihood:

$$L(u_{ij}, 1 \le i < j \le N | x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m) = \prod_{1 \le i < j \le N} \frac{e^{-\theta_{ij}} \theta_{ij}^{u_{ij}}}{u_{ij}!}$$

$$\log(\theta_{ij}) = \beta_0 + \beta_1 \log\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}\right) + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

49

Bayesian Statistical Model

- Likelihood: $\binom{N}{2}$ data points, $3N + 5$ parameters

$$L(u_{ij}, 1 \le i < j \le N | x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m) = \prod_{1 \le i < j \le N} \frac{e^{-\theta_{ij}} \theta_{ij}^{u_{ij}}}{u_{ij}!}$$

$$\log(\theta_{ij}) = \beta_0 + \beta_1 \log\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}\right) + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

50

Bayesian Statistical Model

- Likelihood: $\binom{N}{2}$ data points, $3N + 5$ parameters

$$L(u_{ij}, 1 \le i < j \le N | x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m) = \prod_{1 \le i < j \le N} \frac{e^{-\theta_{ij}} \theta_{ij}^{u_{ij}}}{u_{ij}!}$$

$$\log(\theta_{ij}) = \beta_0 + \beta_1 \log\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}\right) + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

- Posterior distribution

$$\pi(x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m | u_{ij}, 1 \le i < j \le N) \propto L(u_{ij}, 1 \le i < j \le N | x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m) prior$$

51

Statistical Inference

- Algorithm: Bayesian 3D constructor for Hi-C data (BACH)

$$\pi(x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m | u_{ij}, 1 \le i < j \le N)$$

52

Statistical Inference

- Algorithm: Bayesian 3D constructor for Hi-C data (BACH)

$$\pi(x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m | u_{ij}, 1 \le i < j \le N)$$

➢ Initialization 1: use Poisson regression to obtain the initial values for $\beta_0, \beta_e, \beta_g, \beta_m$. Set $\beta_1 = -1$.

$$u_{ij} \sim Poisson(\theta_{ij}) \quad \log(\theta_{ij}) = \beta_0 + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

53

Statistical Inference

- Algorithm: Bayesian 3D constructor for Hi-C data (BACH)

$$\pi(x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m | u_{ij}, 1 \le i < j \le N)$$

➢ Initialization 1: use Poisson regression to obtain the initial values for $\beta_0, \beta_e, \beta_g, \beta_m$. Set $\beta_1 = -1$.

$$u_{ij} \sim Poisson(\theta_{ij}) \quad \log(\theta_{ij}) = \beta_0 + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

➢ Initialization 2: use sequential important sampling to get the initial 3D chromosomal structure $\{x_i, y_i, z_i, 1 \le i \le N\}$.

54

Statistical Inference

- Algorithm: Bayesian 3D constructor for Hi-C data (BACH)

$$\pi(x_i, y_i, z_i, 1 \le i \le N, \beta_0, \beta_1, \beta_e, \beta_g, \beta_m | u_{ij}, 1 \le i < j \le N)$$

- Initialization 1: use Poisson regression to obtain the initial values for $\beta_0, \beta_e, \beta_g, \beta_m$. Set $\beta_1 = -1$.

$$u_{ij} \sim Poisson(\theta_{ij}) \quad \log(\theta_{ij}) = \beta_0 + \beta_e \log(e_i e_j) + \beta_g \log(g_i g_j) + \beta_m \log(m_i m_j)$$

- Initialization 2: use sequential important sampling to get the initial 3D chromosomal structure $\{x_i, y_i, z_i, 1 \le i \le N\}$.

- Refinement: use Gibbs sampler with hybrid Monte Carlo to refine the initial values for parameters.

55

SIS in BACH: Outline

- Goal: use sequential importance sampling to sequentially put *N* loci into 3D space, i.e. sample from:

$$\pi(x_i, y_i, z_i, 1 \le i \le N | u_{ij}, 1 \le i < j \le N)$$

56

SIS in BACH: Outline

- Goal: use sequential importance sampling to sequentially put *N* loci into 3D space, i.e. sample from:

$$\pi(x_i, y_i, z_i, 1 \le i \le N | u_{ij}, 1 \le i < j \le N)$$

- Bridging distributions:

$$\pi_t(x_i, y_i, z_i, 1 \le i \le t | u_{ij}, 1 \le i < j \le t)$$

57

SIS in BACH: Outline

- Goal: use sequential importance sampling to sequentially put *N* loci into 3D space, i.e. sample from:

$$\pi(x_i, y_i, z_i, 1 \le i \le N | u_{ij}, 1 \le i < j \le N)$$

- Bridging distributions:

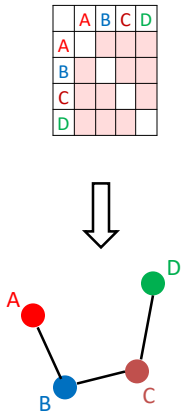
$$\pi_t(x_i, y_i, z_i, 1 \le i \le t | u_{ij}, 1 \le i < j \le t)$$

- Proposal distributions (given the first *t*-1 loci, put the *t* th locus in to 3D space):

$$g_t(x_t, y_t, z_t | x_i, y_i, z_i, 1 \le i \le t-1, u_{ij}, 1 \le i < j \le t)$$

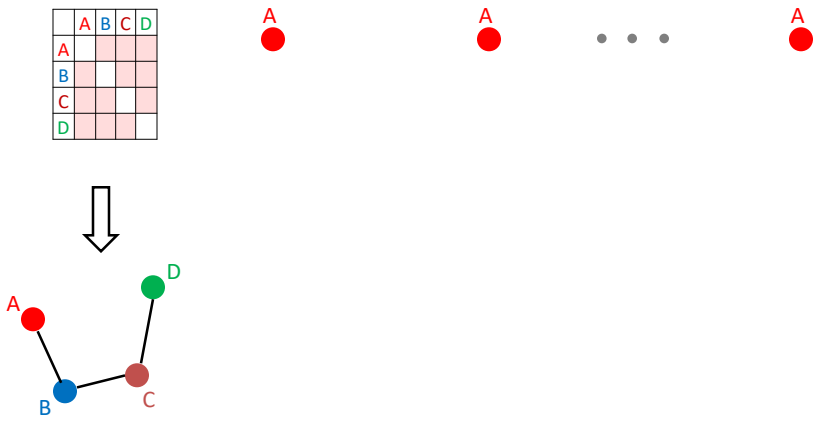
58

SIS in BACH: Illustration



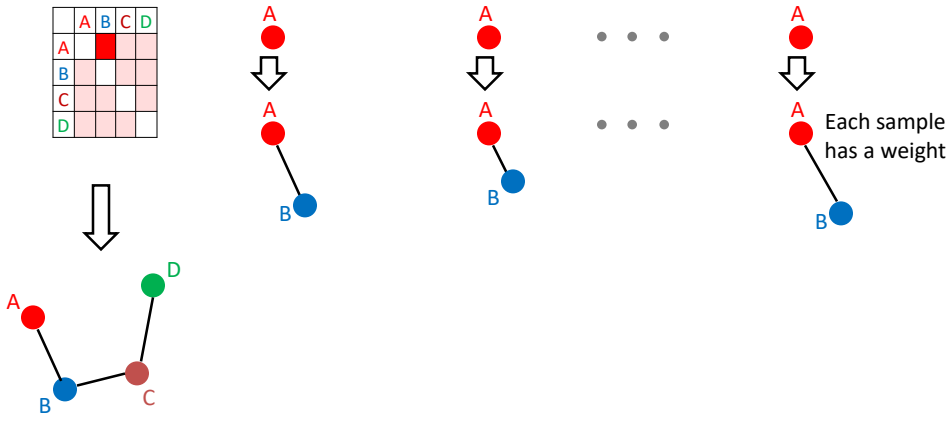
59

SIS in BACH: Illustration



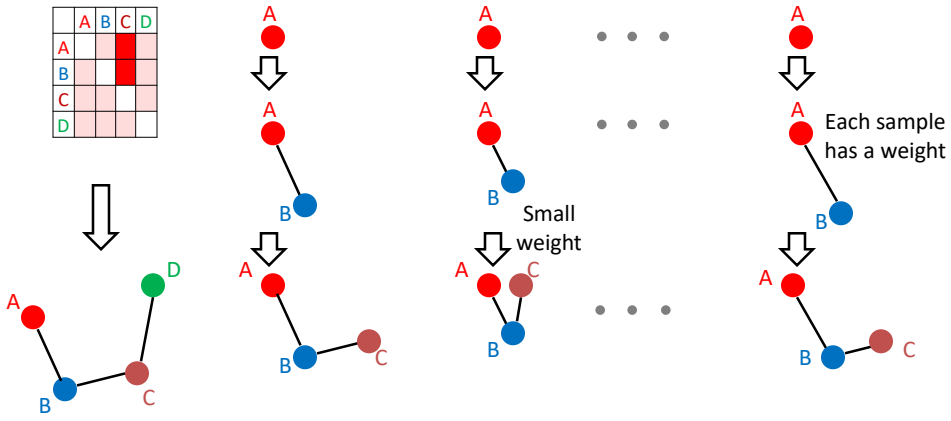
60

SIS in BACH: Illustration



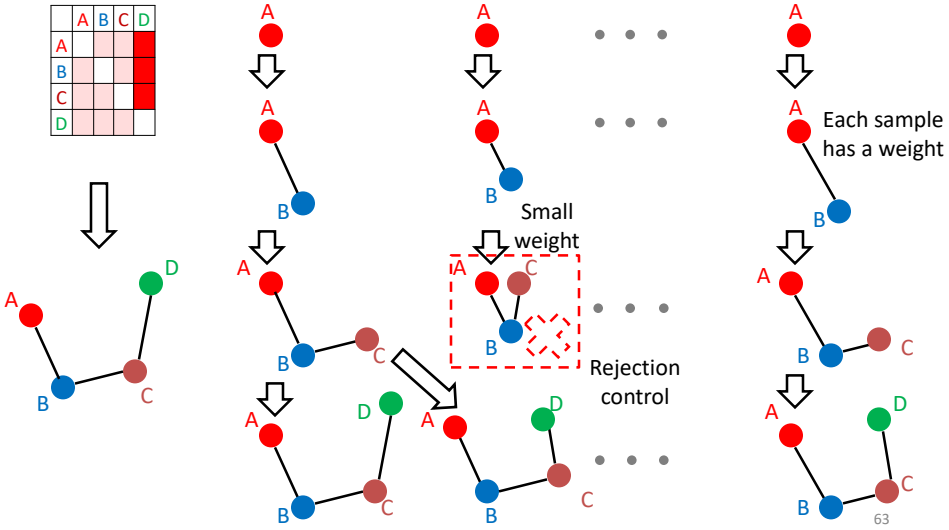
61

SIS in BACH: Illustration



62

SIS in BACH: Illustration



Hybrid Monte Carlo

- Goal: do efficient group move to refine initial 3D chromosomal structure, since local 3D coordinates are highly correlated.

- Combine molecular dynamics with Metropolis acceptance-rejection rule.

64
Duane, et al, 1987

Hybrid Monte Carlo in BACH

- Goal: sampling from

$$\pi(x_i, y_i, z_i, 1 \le i \le N | u_{ij}, 1 \le i < j \le N)$$

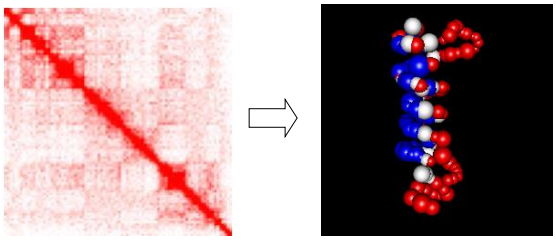
- Take partial derivate of log likelihood over 3D coordinates $(x_i, y_i, z_i, 1 \le i \le N)$.

- Run the leap-frog algorithm, adaptively tune the time interval to achieve acceptance rate ~ 90%.

65

Conclusions

- BACH: reconstruct chromosome 3D structures from Hi-C data
- Remove systematic biases
- Predicted spatial distances are consistent with FISH data
- Elongation of chromatin is highly associated with genetic/epigenetic features.
- Separation of compartments of A and B can be visualized.



66

References

- **Hu M**, Deng K, Qin ZS, Dixon J, Selvaraj S, Fang J, Ren B, Liu JS. (2013) Bayesian inference of three-dimensional chromosomal organization. *PLoS Comput Biol.* **9** e1002893.
  http://www.people.fas.harvard.edu/~junliu/BACH/

- Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, **Hu M**, Liu JS and Ren B. (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* , 485, 376-380.

67