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

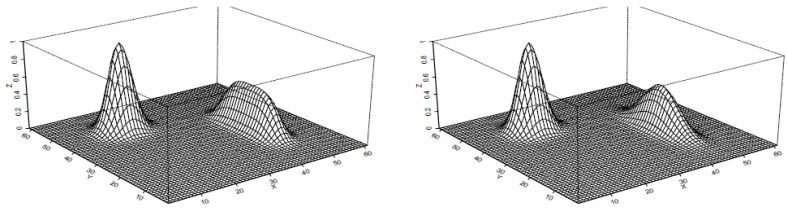
$$E_f[h(X)] = \int_{\mathcal{X}} h(x) \frac{f(x)}{g(x)} g(x) dx$$

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 $\alpha \in E$,

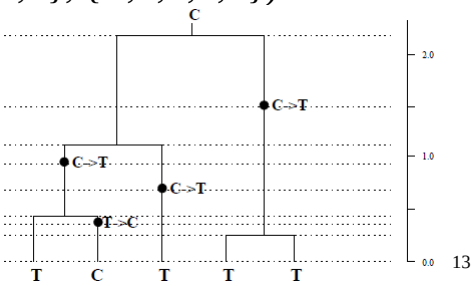
its progeny is $\begin{cases} \alpha & \text{with prob.} 1-\mu, \\ \beta & \text{with prob.} \mu P_{\alpha\beta}. \end{cases}$

12

Example data

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

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

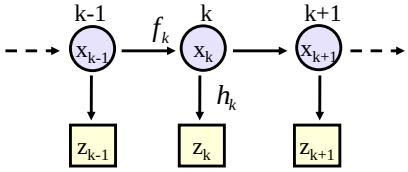


13

Particle filter

Michael Rubinstein
IDC

Dynamic System



Stochastic diffusion

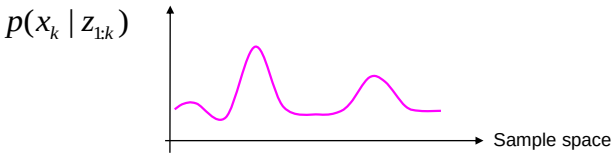
State equation: $x_k = f_k(x_{k-1}, v_k)$

- x_k state vector at time instant k
- f_k state transition function, $f_k : R^{N_x} \times R^{N_v} \to R^{N_x}$
- v_k i.i.d process noise

Observation equation: $z_k = h_k(x_k, w_k)$

- z_k observations at time instant k
- h_k observation function, $h_k : R^{N_x} \times R^{N_w} \to R^{N_z}$
- w_k i.i.d measurement noise

Recursive Bayes filter



- Prediction:

$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \quad (1)$$

- Update:

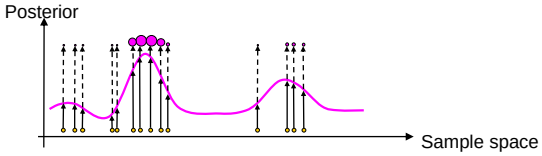
$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \quad (2)$$

$$p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k) p(x_k \mid z_{1:k-1}) dx_k$$

Particle filtering

- Many variations, one general concept:

Represent the posterior pdf by a set of randomly chosen weighted samples (particles)



- Randomly Chosen = Monte Carlo (MC)
- As the number of samples become very large – the characterization becomes an equivalent representation of the true pdf

Particle filtering

- Compared to methods we've mentioned last time
  - Can represent any arbitrary distribution
    - multimodal support
  - Keep track of many hypotheses as there are particles
  - **Approximate representation of complex model rather than exact representation of simplified model**

- The basic building-block: *Importance Sampling*

Monte Carlo integration

- Evaluate complex integrals using probabilistic techniques
- Assume we are trying to estimate a complicated integral of a function *f* over some domain *D*:

$$F = \int_D f(\vec{x}) d\vec{x}$$

- Also assume there exists some PDF *p* defined over *D*

Monte Carlo integration

- Then

$$F = \int_D f(\vec{x}) d\vec{x} = \int_D \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x}) d\vec{x}$$

- But

$$\int_D \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x}) d\vec{x} = E\left[\frac{f(\vec{x})}{p(\vec{x})}\right], x \sim p$$

- This is true for <u>any</u> PDF *p* over D!

Monte Carlo integration

- Now, if we have i.i.d random samples $\vec{x}_1,...,\vec{x}_N$ sampled from p, then we can approximate $E\left[\frac{f(\vec{x})}{p(\vec{x})}\right]$ by

$$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\vec{x}_i)}{p(\vec{x}_i)}$$

- Guaranteed by law of large numbers:

$$N \to \infty, F_N \overset{a.s}{\to} E\left[\frac{f(\vec{x})}{p(\vec{x})}\right] = F$$

Importance Sampling (IS)

- What about $p(\vec{x}) = 0$?
- If p is very small, f / p can be arbitrarily large, 'damaging' the average
  - Design p such that f / p is bounded
  - Rule of thumb: take p similar to f as possible

Importance weights

***Importance* or *proposal* density**

- The effect: get more samples in 'important' areas of f, i.e. where f is large

IS for Bayesian estimation

- We draw the samples from the importance density $q(x_{0:k} \mid z_{1:k})$ with importance weights

$$w_k^i \propto \frac{p(x_{0:k} \mid z_{1:k})}{q(x_{0:k} \mid z_{1:k})}$$

- Sequential update (after some calculation…)

Particle update

$$x_k^i \sim q(x_k \mid x_{k-1}^i, z_k)$$

Weight update

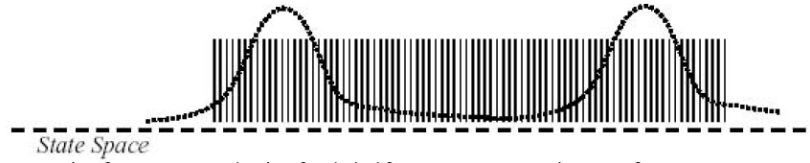
$$w_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)}$$

Sequential Importance Sampling (SIS)

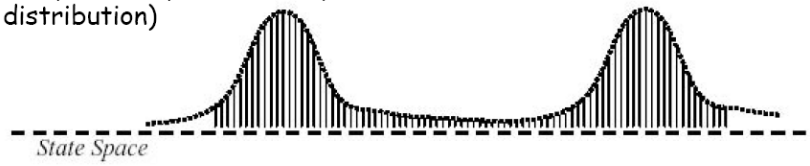
$$\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{SIS}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\right]$$

- FOR i=1:N
  - Draw $x_k^i \sim q(x_k \mid x_{k-1}^i, z_k)$
  - Update weights $w_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)}$
- END
- Normalize weights

Choice of importance density



Sample from prior belief q(x) (for instance, the uniform distribution)



Compute importance weights, w(x) = p(x) /q(x)

Hsiao et al.

Choice of importance density

- Most common (suboptimal): the transitional prior

$$q(x_k \mid x_{k-1}^i, z_k) = p(x_k \mid x_{k-1}^i)$$

$$\Rightarrow w_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} = w_{k-1}^i p(z_k \mid x_k^i)$$

Grid filter weight update: $$w_{k|k}^i = \frac{w_{k|k-1}^i p(z_k \mid x_k^i)}{\sum_{j=1}^{N_s} w_{k|k-1}^j p(z_k \mid x_k^j)}$$

The degeneracy phenomenon

- Unavoidable problem with SIS: after a few iterations most particles have negligible weights
  - Large computational effort for updating particles with very small contribution to $p(x_k \mid z_{1:k})$
- Measure of degeneracy - the effective sample size:

$$\boxed{N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}}$$

  - Uniform: $N_{eff} = N$, severe degeneracy: $N_{eff} = 1$

Resampling

- The idea: when degeneracy is above some threshold, eliminate particles with low importance weights and multiply particles with high importance weights

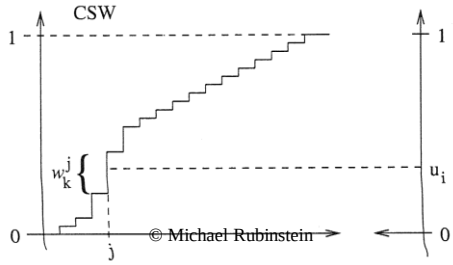
$$\{x_k^i, w_k^i\}_{i=1}^N \rightarrow \{x_k^{i*}, \tfrac{1}{N}\}_{i=1}^N$$

- The new set is generated by sampling with replacement from the discrete representation of $p(x_k \mid z_{1:k})$ such that $\Pr\{x_k^{i*} = x_k^j\} = w_k^j$

Resampling

$\left[\{x_k^{i*}, w_k^i\}_{i=1}^N\right] = \text{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^N\right]$

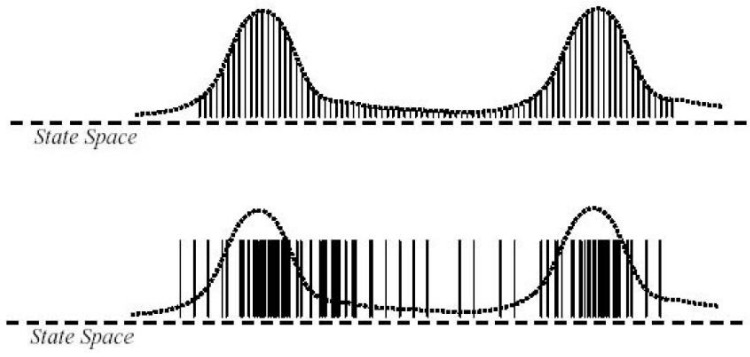
- Generate N i.i.d variables $u_i \sim U[0,1]$
- Sort them in ascending order
- Compare them with the cumulative sum of normalized weights



Ristic et al.

Resampling

- Complexity: O(NlogN)
  - O(N) sampling algorithms exist



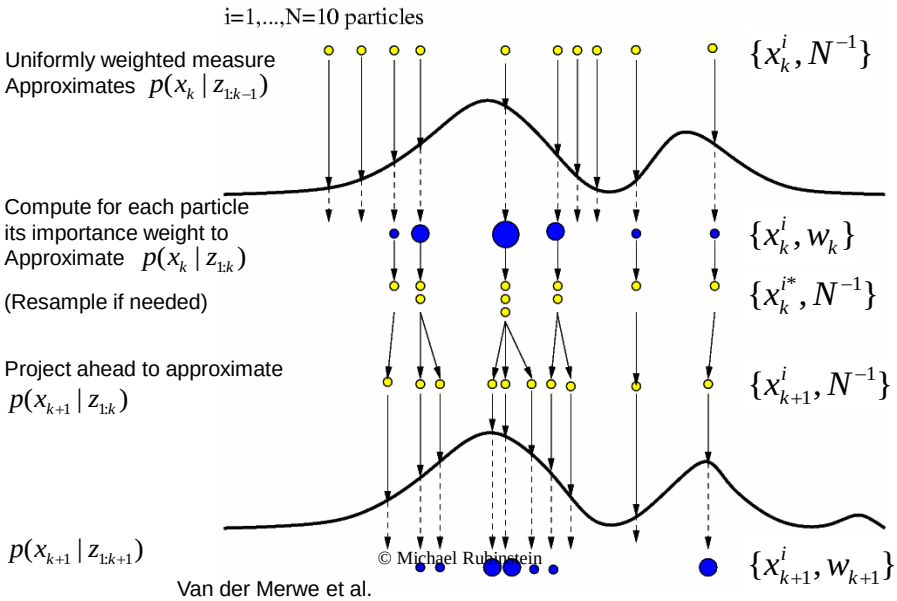
Hsiao et al.

Generic PF

$$\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{PF}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\right]$$

- Apply SIS filtering $\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{SIS}\left[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k\right]$
- Calculate N_{eff}
- IF $N_{eff} < N_{thr}$
  - $\left[\{x_k^i, w_k^i\}_{i=1}^N\right] = \mathrm{RESAMPLE}\left[\{x_k^i, w_k^i\}_{i=1}^N\right]$
- END

Generic PF



Van der Merwe et al.

PF variants

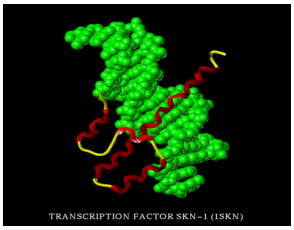
- Sampling Importance Resampling (SIR)
- Auxiliary Sampling Importance Resampling (ASIR)
- Regularized Particle Filter (RPF)
- Local-linearization particle filters
- Multiple models particle filters (maneuvering targets)
- …

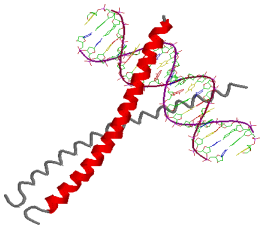
Sampling Importance Resampling (SIR)

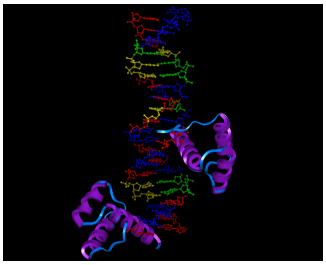
- A.K.A Bootstrap filter, Condensation

- **Initialize** $\{x_0^i, w_0^i\}_{i=1}^N$ from prior distribution X_0
- For k > 0 do
  - **Resample** $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ into $\{x_{k-1}^{i^*}, \frac{1}{N}\}_{i=1}^N$
  - **Predict** $x_k^i \sim p(x_k \mid x_{k-1} = x_{k-1}^{i^*})$
  - **Reweight** $w_k^i = p(z_k \mid x_k = x_k^i)$
  - **Normalize** weights
  - **Estimate** $\hat{x}_k$ (for display)

Appliation: Transcription Factor Binding Sites Discovery







35

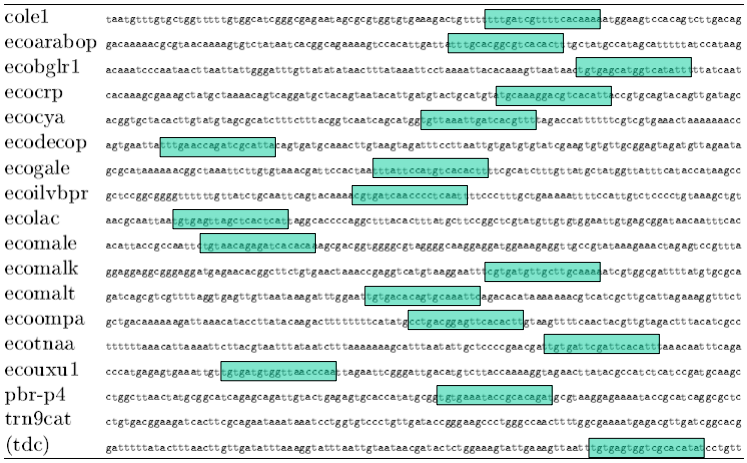
Example: cyclic receptor protein (CRP)

cole1	taatgtttgtgctggttttttgtggcatcgggcgagaatagcgcgtggtgtgaaagactgtttttttgatcgttttcacaaaaatggaagtccacagtcttgacag
ecoarabop	gacaaaaacgcgtaacaaaagtgtctataatcacggcagaaaagtccacattgattatttgcacggcgtcacactttgctatgccatagcatttttatccataag
ecobglr1	acaaatcccaataacttaattattgggatttgttatatataactttataaatttcctaaaattacacaaagttaataactgtgagcatggtcatatttttatcaat
ecocrp	cacaaagcgaaagctatgctaaaacagtcaggatgctacagtaatacattgatgtactgcatgtatgcaaaggacgtcacattaccgtgcagtacagttgatagc
ecocya	acggtgctacacttgtatgtagcgcatctttctttacggtcaatcagcatggtgtttaaattgatcacgttttagaccatttttcgtcgtgaaactaaaaaaacc
ecodecop	agtgaattatttgaaccagatcgcattacagtgatgcaaacttgtaagtagatttccttaattgtgatgtgtatcgaagtgtgttgcggagtagatgttagaata
ecogale	gcgcataaaaaacggctaaaattcttgtgtaaacgattccactaatttattccatgtcacactttcgcatctttgttatgctatggttatttcataccataagcc
ecoilvbpr	gctccggcggggttttttgttatctgcaattcagtacaaaacgtgatcaacccctcaattttccctttgctgaaaaattttccattgtctccccgtaaagctgt
ecolac	aacgcaattaatgtgagttagctcactcattaggcaccccaggctttacactttatgcttccggctcgtatgttgtgtggaattgtgagcggataacaatttcac
ecomale	acattaccgccaattctgtaacagagatcacacaaagcgacggtgggggcgtagggggcaaggaggatggaaagaggttgccgtataaagaaactagagtccgtttta
ecomalk	ggaggaggcgggaggatgagaacacggcttctgtgaactaaaccgaggtcatgtaaggaatttcgtgatgttgcttgcaaaaatcgtggcgattttatgtgcgca
ecomalt	gatcagcgtcgttttaggtgagttgttaataaagattggaattgtgacacagtgcaaattcagacacatataaaaaacgtcatcgcttgcattagaaaggtttct
ecoompa	gctgacaaaaaagattaaacatacctttatacaagactttttttcatatgcctgacggagttcacacttgtaagttttcaactacgttgtagacttttacatcgcc
ecotnaa	ttttttaaacatttaaaattctttacgtaatttataatctttaaaaaaagcatttaatattgctcccccgaacgattgtgattcgattcacatttaaacaatttcaga
ecouxu1	cccatgagagtgaaattgttgtgatgtggttcaccccaattagaattcgggattgacatgtcttaccaaaaggtagaacttatacgccatctcatcccgatgcaagc
pbr-p4	ctggcttaactatgcggcatcagagcagattgtactgagagtgcaccatatgcggtgtgaaataccgcacagatgcgtaaggagaaaataccgcatcaggcgctc
trn9cat	ctgtgacggaagatcacttcgcagaataaataaatcctggtgtccctgttgataccgggaagccctgggccaacttttggcgaaaatgagacgttgatcggcacg
(tdc)	gatttttatactttaacttgttgatatttaaaggtatttaattgtaataacgatactctggaaagtattgaaagttaatttgtgagtggtcgcacatatcctgtt

36

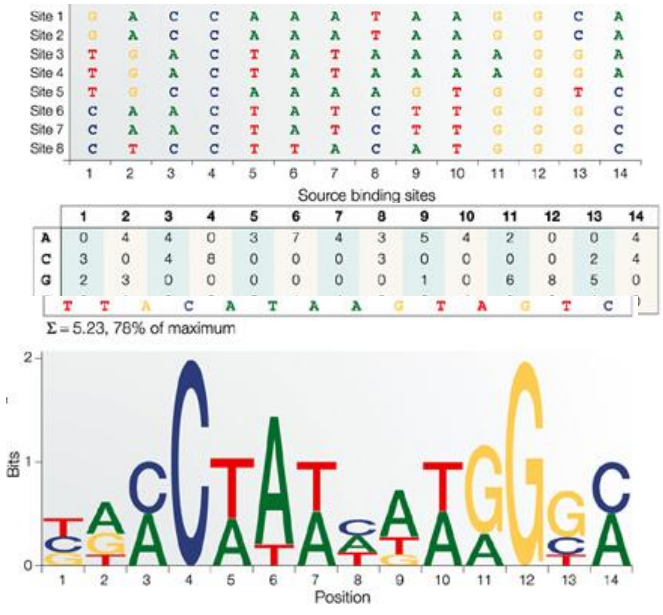
Stormo and Hartzell, 1989

Example: cyclic receptor protein (CRP)



Stormo and Hartzell, 1989

Transcription factor binding site (TFBS)



Motif identification model

a_1
aaaggtcgagtagctactcgatcgatactagcaatcgttaccctagctcgatcgaaa
a_2
acgtgagatcagctatgaccgatagctactcgataaccg
a_3
gaatagctactcgatcgatactagcaatcgttaccctagctcgatcgagatggaaag
• • •
a_J
acgtgagatcagctatcgatcgattgataactactcgtacgtat

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

39

Posterior distributions

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

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

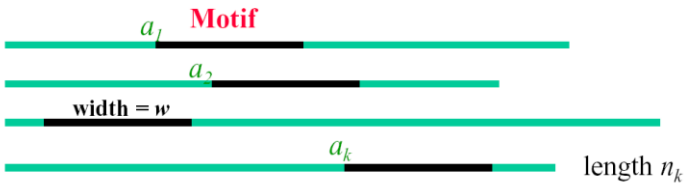
DNA sequence data

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

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

40

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 $\boldsymbol{p_i}=(p_{i,1},\ldots, p_{i,20})$

41

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 $P(\theta \mid \pi, seq, \theta_0)$

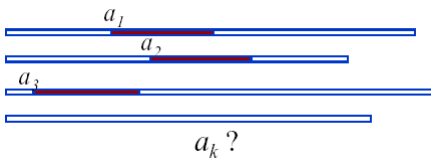
42

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.

43

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

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

44

References

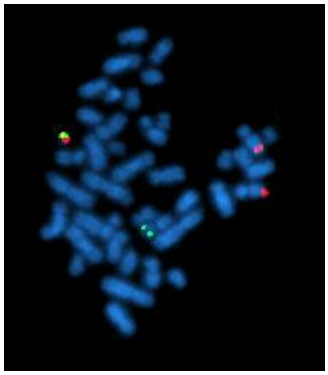
- Lawrence et al. (1993) *Science.*
- Liu, Neuwald and Lawrence (1995) *JASA.*
- Liu and Lawrence (1999) *Bioinformatics.*

45

Infer the 3D shape of chromosomes

Microscopic Methods

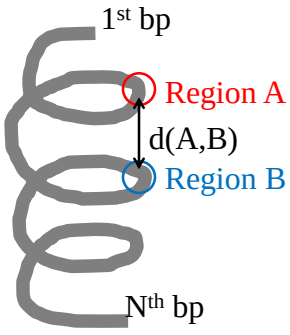
- Fluorescent *in situ* hybridization (FISH)



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

47

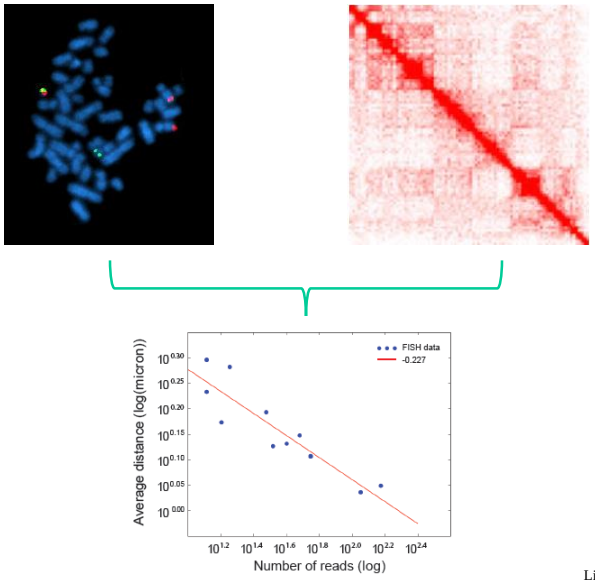
FISH Data Representation



3D chromosomal structure

48

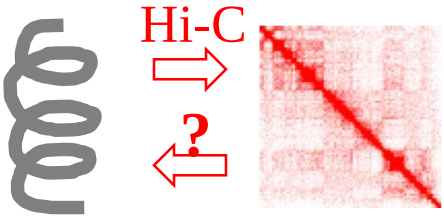
Contact Frequency vs. Spatial Distance



49

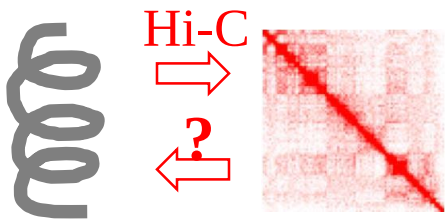
Lieberman-Aiden, et al, 2009

Problem setting



50

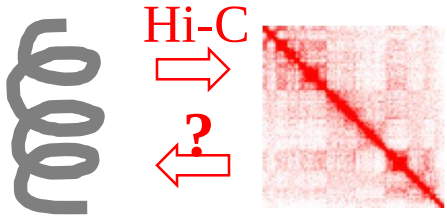
Problem setting



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

51

Problem setting



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

52

Yaffe and Tanay, 2011

Beads-on-a-string Representation

ACGTAGCTAGATACTGTAGTGTAGTTTGGAACCTGAGGG

53

Beads-on-a-string Representation

ACGTAGCTAGATACTGTAGTGTAGTTTGGAACCTGAGGG

54

Beads-on-a-string Representation

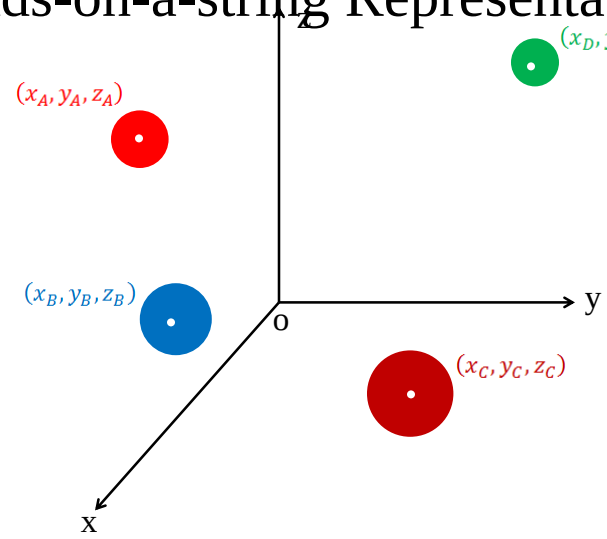
ACGTAGCTAG ATACTGTAGT GTAGTTTGGA ACCTGAGGG

55

Beads-on-a-string Representation

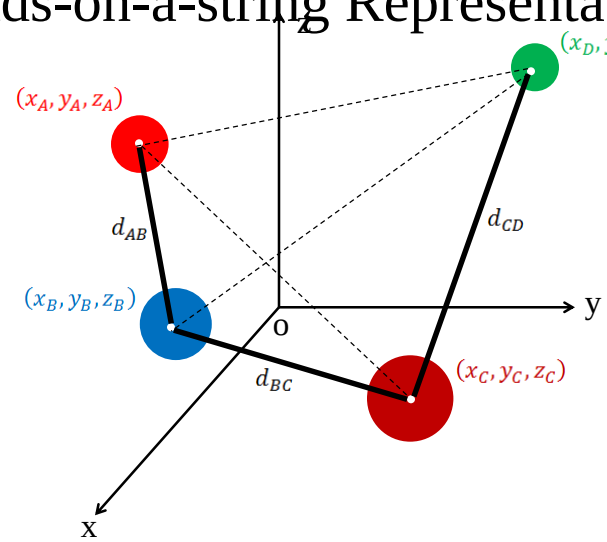
56

Beads-on-a-string Representation



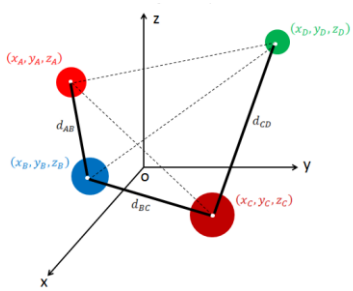
57

Beads-on-a-string Representation



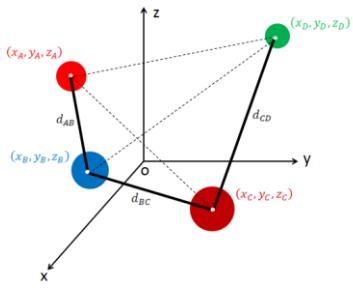
58

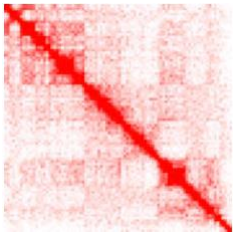
Bayesian Statistical Model



59

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

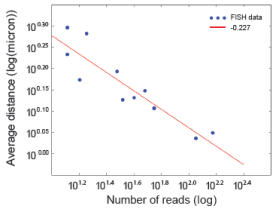
g_i : GC content of locus i

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



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

60

Lieberman-Aiden, et al, 2009

Bayesian Statistical Model

- Likelihood:

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

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

61

Bayesian Statistical Model

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

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

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

62

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

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

63

Statistical Inference

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

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

64

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 $\beta_1 = -1$.

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

65

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 $\{x_i, y_i, z_i, 1 \le i \le N\}$.

66

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.

67

SIS in BACH: Outline

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

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

68

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:

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

69

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:

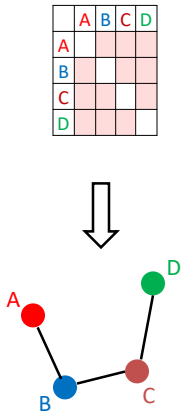
$$\pi_t(x_i, y_i, z_i, 1 \le i \le t | u_{ij}, 1 \le i < j \le t)$$

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

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

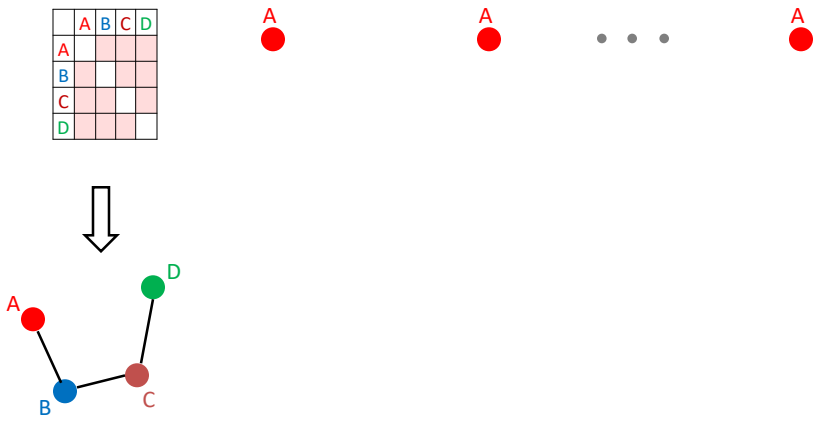
70

SIS in BACH: Illustration



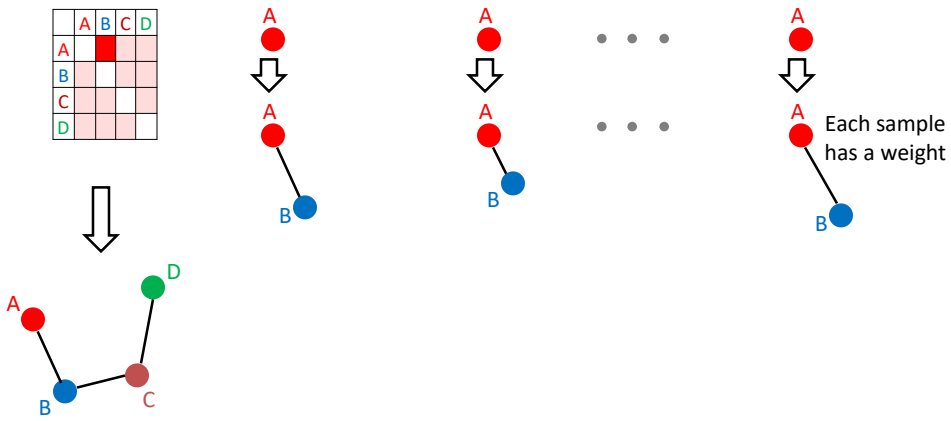
71

SIS in BACH: Illustration



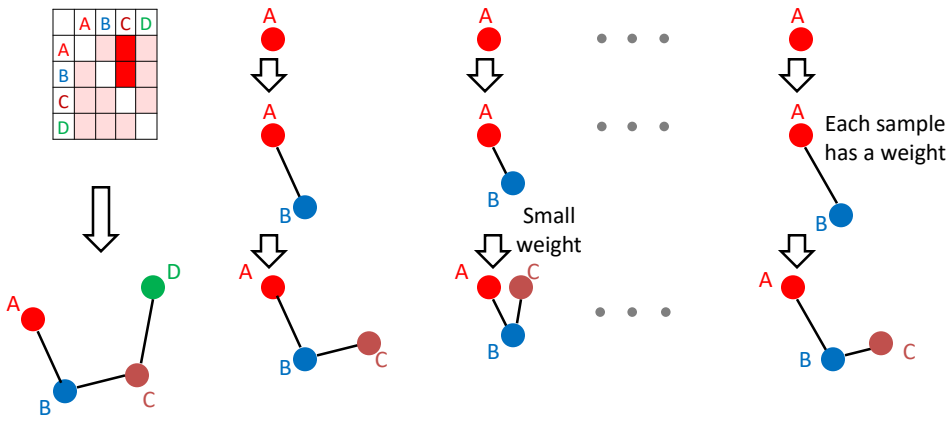
72

SIS in BACH: Illustration



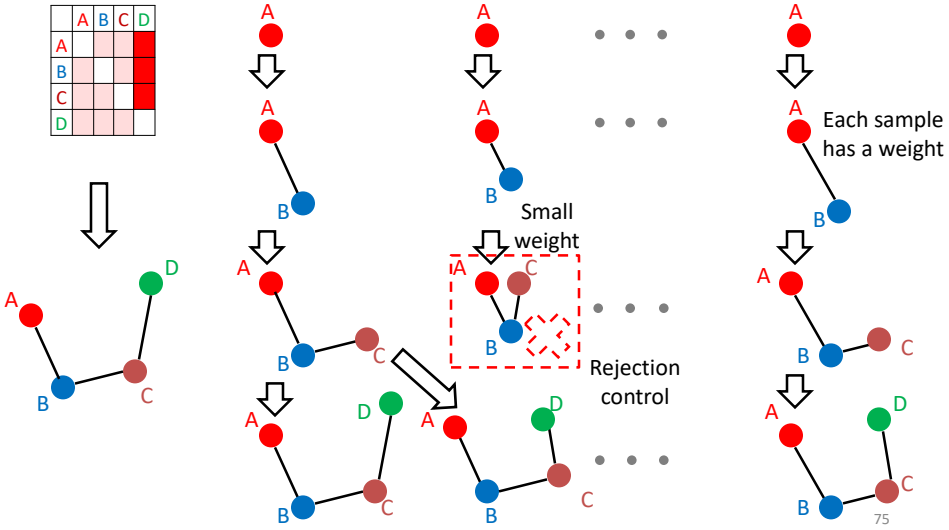
73

SIS in BACH: Illustration



74

SIS in BACH: Illustration



75

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.

76

Duane, et al, 1987

Hybrid Monte Carlo in BACH

- Goal: sampling from

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

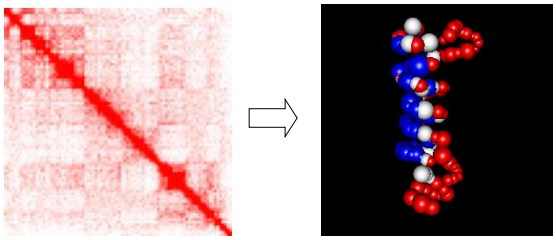
- Take partial derivate of log likelihood over 3D coordinates $(x_i, y_i, z_i, 1 \le i \le N)$.

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

77

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.



78

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.

79