Advanced Statistical Computing

Fall 2018

Steve Qin

Review

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

Importance sampling

- *Importance sampling:*
  to evaluate $E_f[h(X)] = \int_{\aleph} h(x) f(x) dx$
  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

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

Sequential importance sampling

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

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

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

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

then we have

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

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 $\pi(x_t \mid \mathbf{x_{t\text{-}1}})$ in designing $g_t(x_t \mid \mathbf{x_{t\text{-}1}})$
- However, the scheme is impractical since it requires the knowledge of marginal distribution $\pi(\boldsymbol{x}_t)$.

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,…,d -1 and $\pi_d = \pi$.
- Note the π_d are only required to be known up to a normalizing constant.

The SIS procedure

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

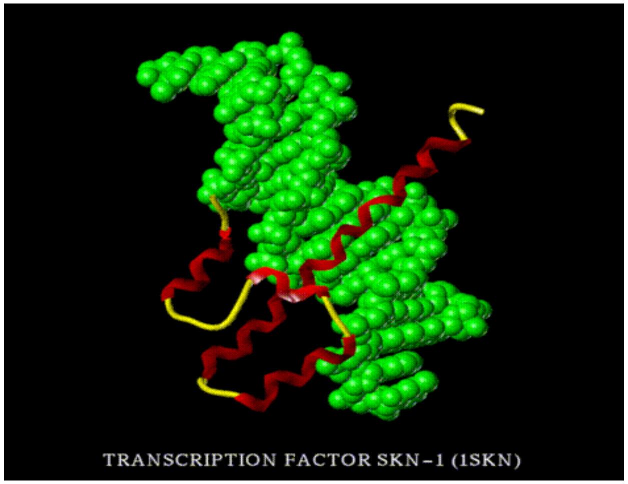
- Draw $X_t = x_t$ from $g_t(x_t \mid x_{t-1})$, and let

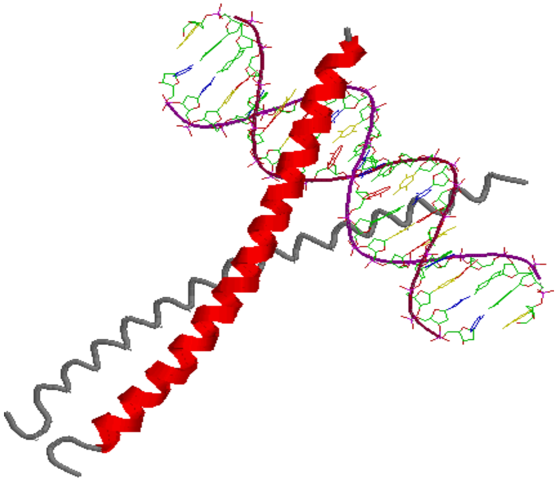
  $\mathbf{x_t} = (\mathbf{x_{t-1}}, x_t)$
- Compute $$u_t = \frac{\pi_t(\mathbf{x_t})}{\pi_{t-1}(\mathbf{x_{t-1}})g_t(x_t \mid \mathbf{x_{t-1}})}$$

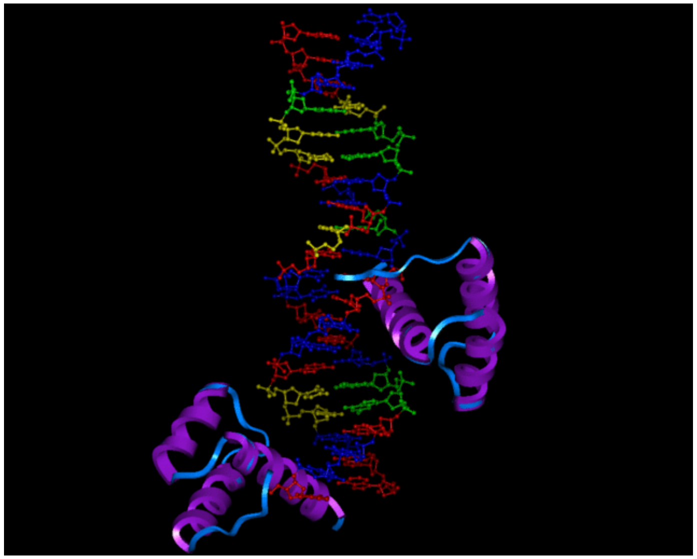
  and let $w_t = w_{t-1}\, u_t$
- u_t : incremental weight.
- The key idea is to breaks a difficult task into manageable pieces.
- If w_t is getting too small, reject.

Applications of MCMC and SMC

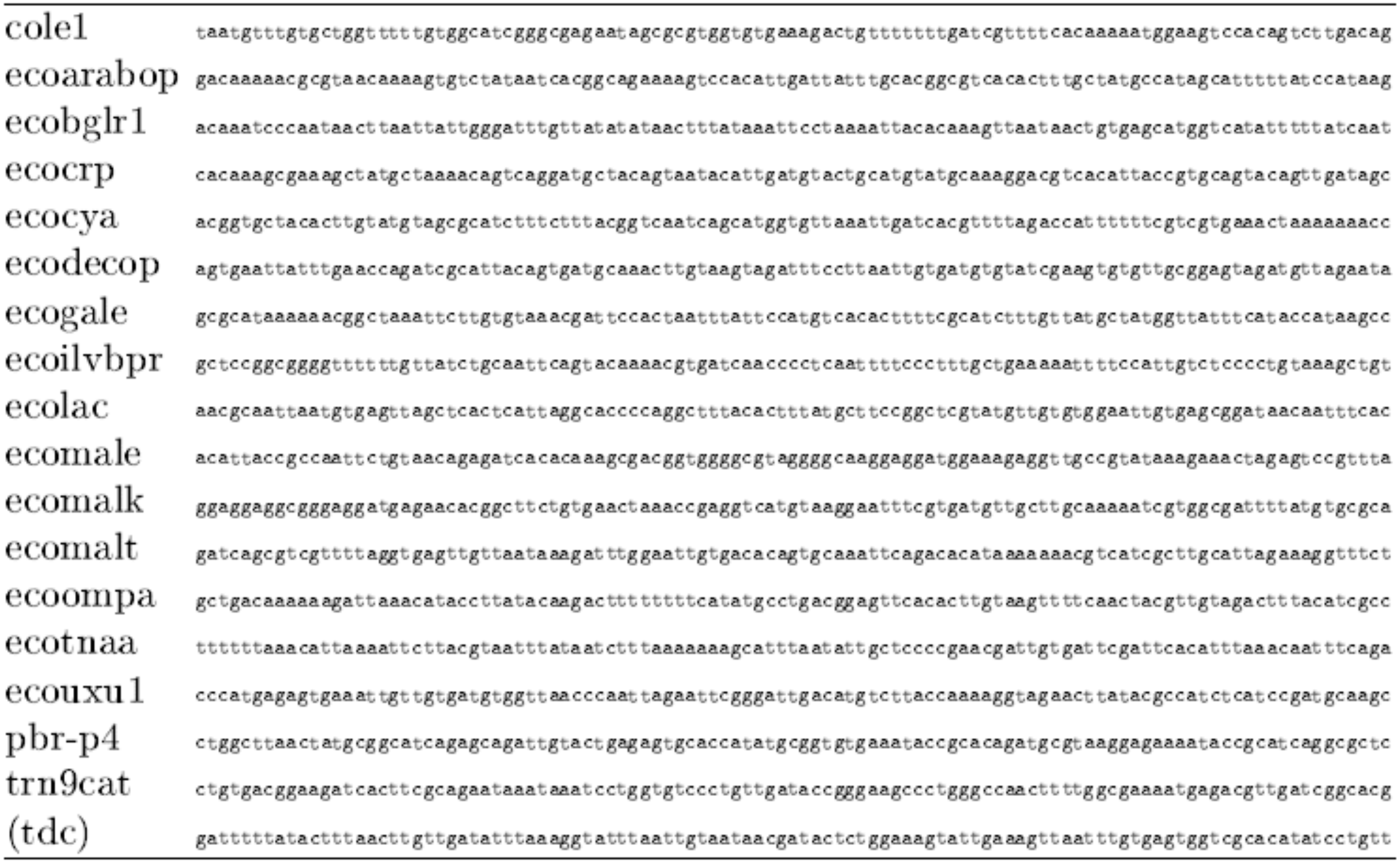
Appliation: Transcription Factor Binding Sites Discovery







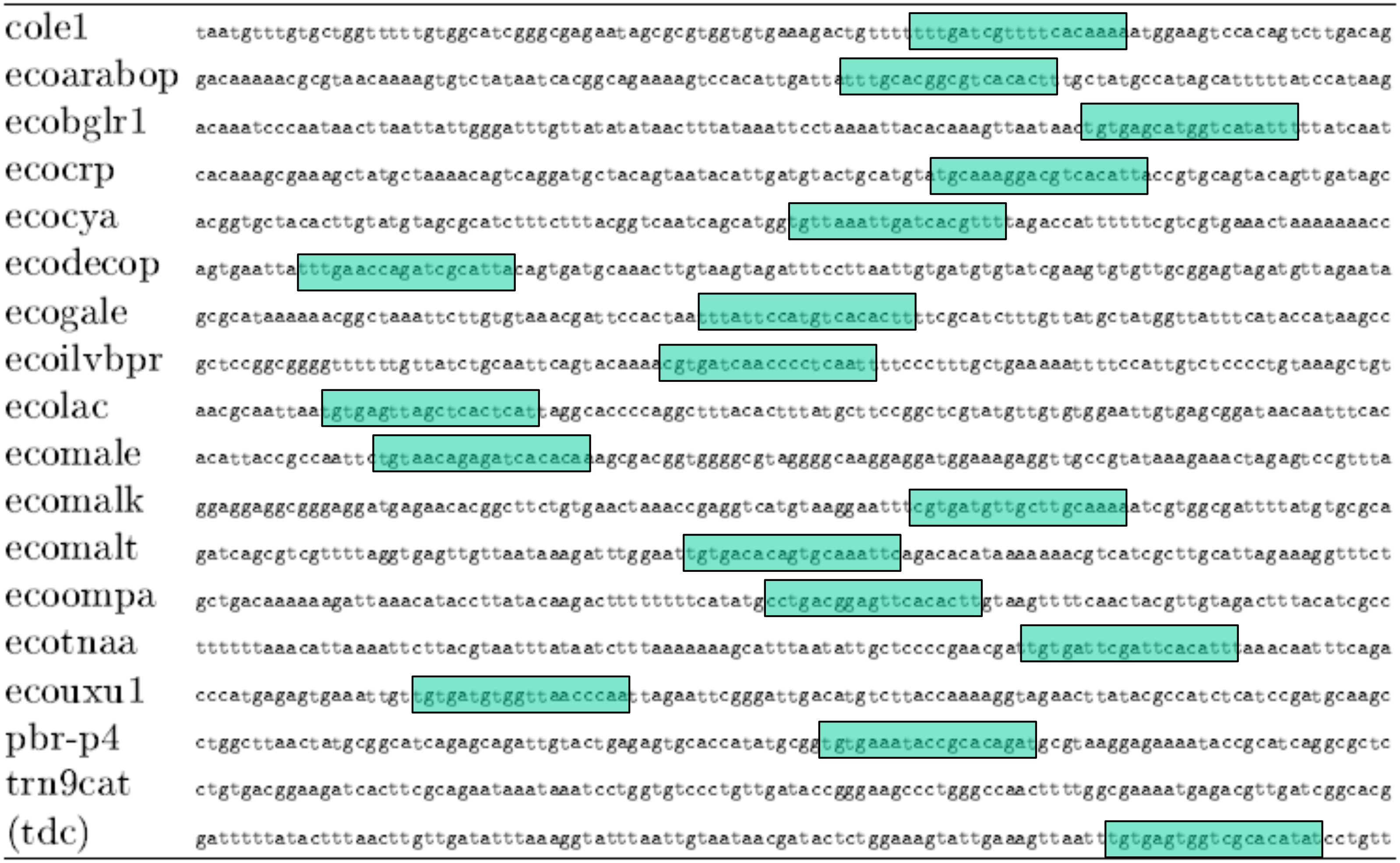
Example: cyclic receptor protein (CRP)



cole1	taatgtttgtgctggtttttgtggcatcgggcgagaatagcgcgtggtgtgaaagactgtttttttgatcgttttcacaaaaatggaagtccacagtcttgacag
ecoarabop	gacaaaaacgcgtaacaaaagtgtctataatcacggcagaaaagtccacattgattatttgcacggcgtcacactttgctatgccatagcatttttatccataag
ecobglr1	acaaatcccaataacttaattattgggatttgttatatataactttataaattcctaaaattacacaaagttaataactgtgagcatggtcatatttttatcaat
ecocrp	cacaaagcgaaagctatgctaaaacagtcaggatgctacagtaatacattgatgtactgcatgtatgcaaaggacgtcacattaccgtgcagtacagttgatagc
ecocya	acggtgctacacttgtatgtagcgcatctttctttacggtcaatcagcatggtgttaaattgatcacgttttagaccattttttcgtcgtgaaactaaaaaaacc
ecodecop	agtgaattatttgaaccagatcgcattacagtgatgcaaacttgtaagtagatttccttaattgtgatgtgtatcgaagtgtgttgcggagtagatgttagaata
ecogale	gcgcataaaaaacggctaaattcttgtgtaaacgattccactaatttattccatgtcacacttttcgcatctttgttatgctatggttatttcataccataagcc
ecoilvbpr	gctccggcggggttttttgttatctgcaattcagtacaaaacgtgatcaacccctcaattttccctttgctgaaaaattttccattgtctcccctgtaaagctgt
ecolac	aacgcaattaatgtgagttagctcactcattaggcaccccaggctttacactttatgcttccggctcgtatgttgtgtggaattgtgagcggataacaatttcac
ecomale	acattaccgccaattctgtaacagagatcacacaaagcgacggtggggcgtaggggcaaggaggatggaaagaggttgccgtataaagaaactagagtccgttta
ecomalk	ggaggaggcgggaggatgagaacacggcttctgtgaactaaaccgaggtcatgtaaggaatttcgtgatgttgcttgcaaaaatcgtggcgattttatgtgcgca
ecomalt	gatcagcgtcgttttaggtgagttgttaataaagatttggaattgtgacacagtgcaaattcagacacataaaaaaacgtcatcgcttgcattagaaaggtttct
ecoompa	gctgacaaaaaagattaaacataccttatacaagactttttttttcatatgcctgacggagttcacacttgtaagttttcaactacgttgtagactttacatcgcc
ecotnaa	ttttttaaacattaaaattcttacgtaatttataatctttaaaaaaagcatttaatattgctccccgaacgattgtgattcgattcacatttaaacaatttcaga
ecouxu1	cccatgagagtgaaattgttgtgatgtggttaacccaattagaattcgggattgacatgtcttaccaaaaggtagaacttatacgccatctcatccgatgcaagc
pbr-p4	ctggcttaactatgcggcatcagagcagattgtactgagagtgcaccatatgcggtgtgaaataccgcacagatgcgtaaggagaaaataccgcatcaggcgctc
trn9cat	ctgtgacggaagatcacttcgcagaataaataaatcctggtgtccctgttgataccgggaagccctgggccaacttttggcgaaaatgagacgttgatcggcacg
(tdc)	gatttttatactttaacttgttgatatttaaaggtatttaattgtaataacgatactctggaaagtattgaaagttaatttgtgagtggtcgcacatatcctgtt

Stormo and Hartzell,

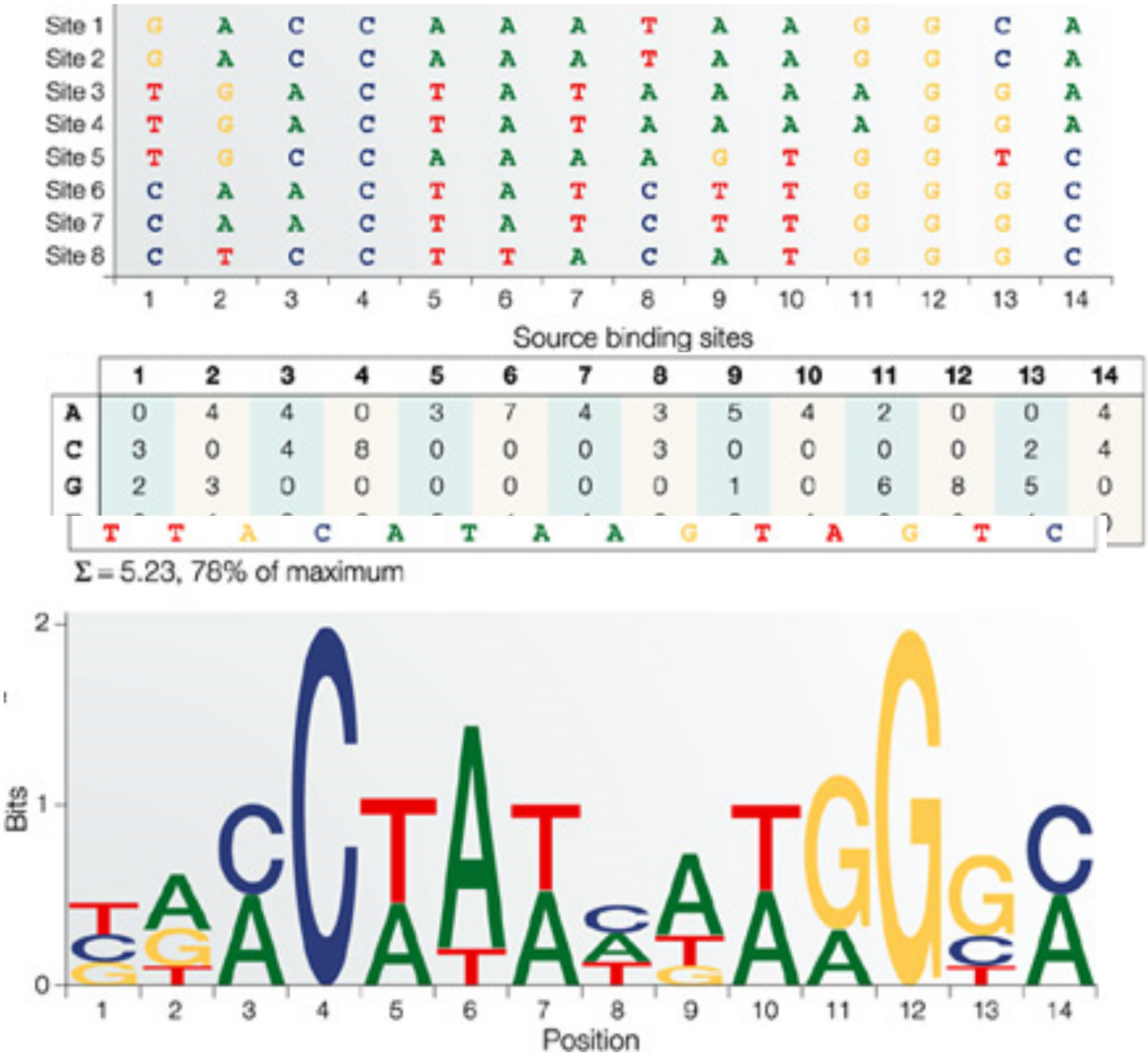
Example: cyclic receptor protein (CRP)



cole1	taatgtttgtgctggtttttgtggcatcgggcgagaatagcgcgtggtgtgaaagactgtttt**tttgatcgttttcacaaaa**atggaagtccacagtcttgacag
ecoarabop	gacaaaaacgcgtaacaaaagtgtctataatcacggcagaaaagtccacattgatta**tttgcacggcgtcacactt**tgctatgccatagcatttttatccataag
ecobglr1	acaaatcccaataacttaattattgggatttgttatatataactttataaattcctaaaattacacaaagttaataac**tgtgagcatggtcatattt**ttatcaat
ecocrp	cacaaagcgaaagctatgctaaaacagtcaggatgctacagtaatacattgatgtactgcatgta**tgcaaaggacgtcacatta**ccgtgcagtacagttgatagc
ecocya	acggtgctacacttgtatgtagcgcatctttctttacggtcaatcagcatgg**tgttaaattgatcacgttt**tagaccattttttcgtcgtgaaactaaaaaaacc
ecodecop	agtgaatta**tttgaaccagatcgcatta**cagtgatgcaaacttgtaagtagatttccttaattgtgatgtgtatcgaagtgtgttgcggagtagatgttagaata
ecogale	gcgcataaaaaacggctaaattcttgtgtaaacgattccactaa**tttattccatgtcacactt**ttcgcatctttgttatgctatggttatttcataccataagcc
ecoilvbpr	gctccggcggggttttttgttatctgcaattcagtacaaaa**cgtgatcaacccctcaatt**ttccctttgctgaaaaattttccattgtctcccctgtaaagctgt
ecolac	aacgcaattaa**tgtgagttagctcactcat**taggcaccccaggctttacactttatgcttccggctcgtatgttgtgtggaattgtgagcggataacaatttcac
ecomale	acattaccgccaattc**tgtaacagagatcacacaa**agcgacggtggggcgtaggggcaaggaggatggaaagaggttgccgtataaagaaactagagtccgttta
ecomalk	ggaggaggcgggaggatgagaacacggcttctgtgaactaaaccgaggtcatgtaaggaattt**cgtgatgttgcttgcaaaa**atcgtggcgattttatgtgcgca
ecomalt	gatcagcgtcgttttaggtgagttgttaataaagatttggaat**tgtgacacagtgcaaattc**agacacataaaaaaacgtcatcgcttgcattagaaaggtttct
ecoompa	gctgacaaaaaagattaaacataccttatacaagacttttttttcatatg**cctgacggagttcacactt**gtaagttttcaactacgttgtagactttacatcgcc
ecotnaa	ttttttaaacattaaaattcttacgtaatttataatctttaaaaaaagcatttaatattgctccccgaacgat**tgtgattcgattcacattt**aaacaatttcaga
ecouxu1	cccatgagagtgaaattgt**tgtgatgtggttaacccaa**ttagaattcgggattgacatgtcttaccaaaaggtagaacttatacgccatctcatccgatgcaagc
pbr-p4	ctggcttaactatgcggcatcagagcagattgtactgagagtgcaccatatgcgg**tgtgaaataccgcacagat**gcgtaaggagaaaataccgcatcaggcgctc
trn9cat	ctgtgacggaagatcacttcgcagaataaataaatcctggtgtccctgttgataccgggaagccctgggccaacttttggcgaaaatgagacgttgatcggcacg
(tdc)	gatttttatactttaacttgttgatatttaaaggtatttaattgtaataacgatactctggaaagtattgaaagttaatt**tgtgagtggtcgcacatat**cctgtt

Stormo and Hartzell, 1989

Transcription factor binding site (TFBS)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Site 1	G	A	C	C	A	A	A	T	A	A	G	G	C	A
Site 2	G	A	C	C	A	A	A	T	A	A	G	G	C	A
Site 3	T	G	A	C	T	A	T	A	A	A	A	G	G	A
Site 4	T	G	A	C	T	A	T	A	A	A	A	G	G	A
Site 5	T	G	C	C	A	A	A	A	G	T	G	G	T	C
Site 6	C	A	A	C	T	A	T	C	T	T	G	G	G	C
Site 7	C	A	A	C	T	A	T	C	T	T	G	G	G	C
Site 8	C	T	C	C	T	T	A	C	A	T	G	G	G	C

Source binding sites

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0	4	4	0	3	7	4	3	5	4	2	0	0	4
C	3	0	4	8	0	0	0	3	0	0	0	0	2	4
G	2	3	0	0	0	0	0	0	1	0	6	8	5	0
T	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

T T A C A T A A G T A G T C

Σ = 5.23, 78% of maximum

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

Motif identification model

a_1

aaaggtcgagtagctactcgatcgatactagcaatcgttaccctagctcgatcgaaa

a_2

acgtgagatcagctatgaccgatagctactcgataaccg

a_3

gaatagctactcgatcgatactagcaatcgttaccctagctcgatcgagatggaaag

• • •

a_J

acgtgagatcagctatcgatcgattgataactactcgtacgtat

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

Posterior distributions

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

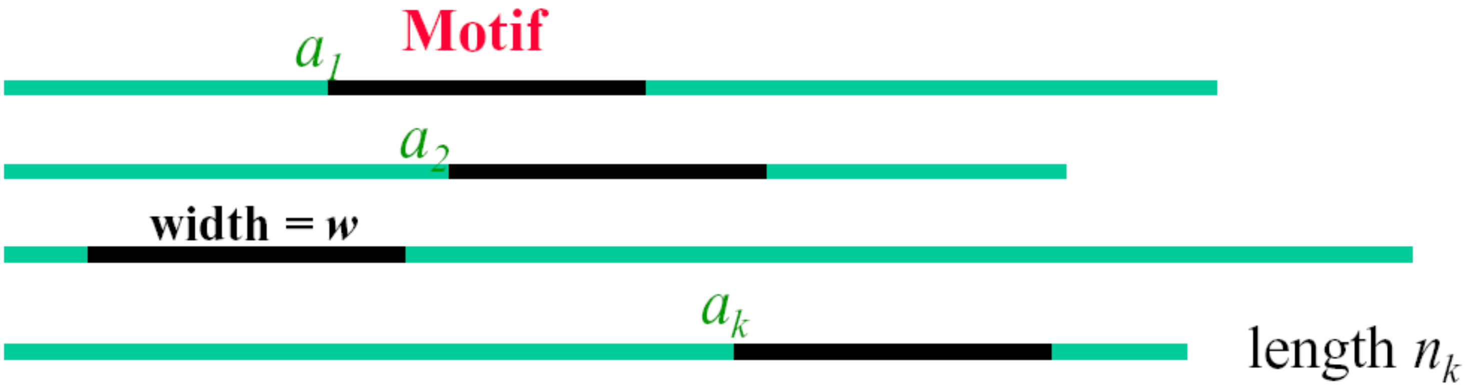
$$p(a_j = l \mid \boldsymbol{\theta_0}, \boldsymbol{\Theta}, \boldsymbol{R_j}, \boldsymbol{A_{-j}}) \propto \prod_{k=1}^{4} \theta_{0k}^{h_k(\boldsymbol{R_j})} \prod_{i=1}^{w} \prod_{k=1}^{4} \left(\frac{\theta_{ik}}{\theta_{0k}} \right)^{h_k(r_{j,l+i-1})} \propto \prod_{i=1}^{w} \prod_{k=1}^{4} \left(\frac{\theta_{ik}}{\theta_{0k}}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

DNA sequence data

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

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

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 $\boldsymbol{i}$ in the motif element follows probability distribution $\boldsymbol{p_i}=(p_{i,1}, \ldots, p_{i,20})$

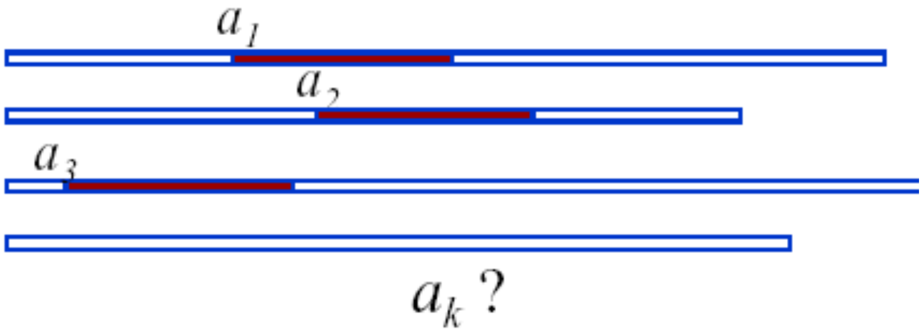
Statistical Model

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

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.

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.

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

- Sample from the predictive distribution of a_k

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

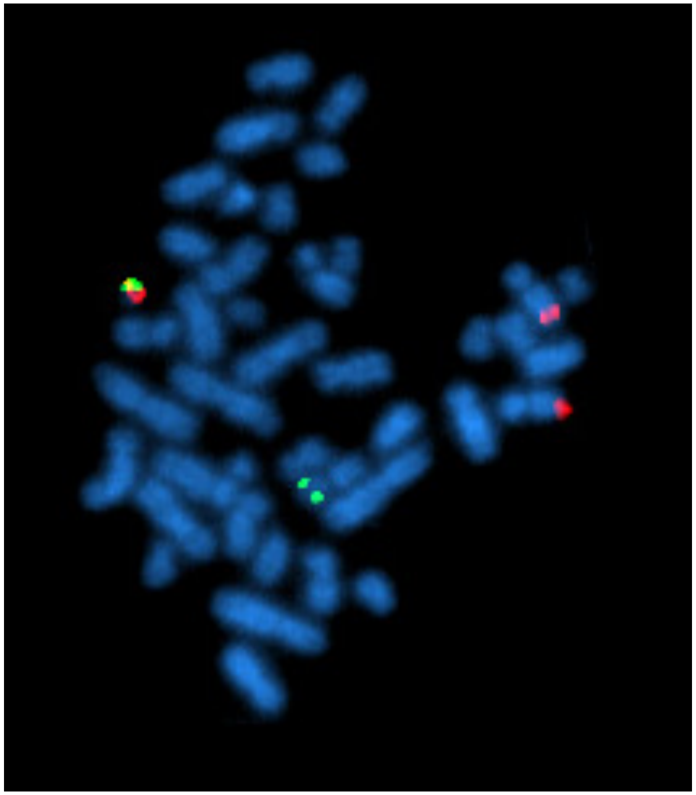
References

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

Infer the 3D shape of chromosomes

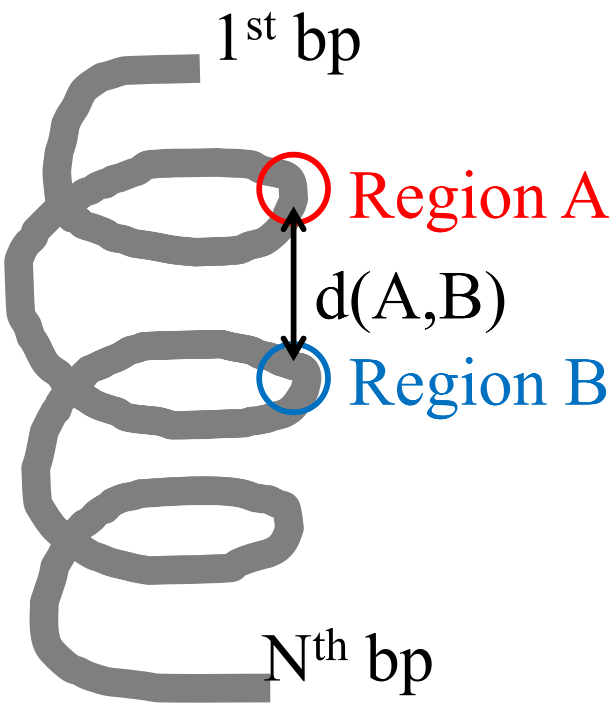
Microscopic Methods

- Fluorescent *in situ* hybridization (FISH)



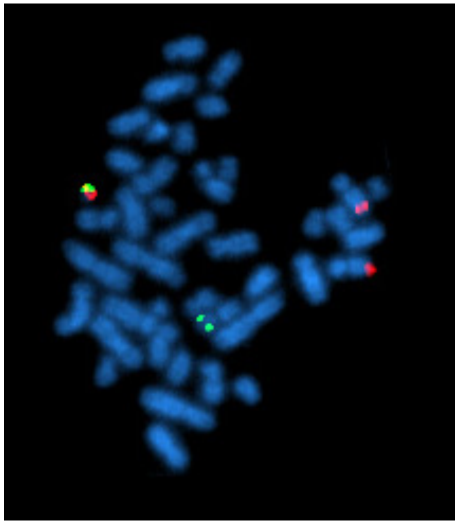
http://en.wikipedia.org/wiki/Cytogenetics

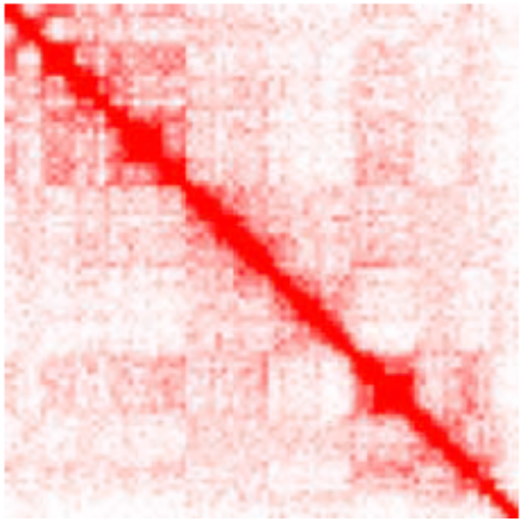
FISH Data Representation

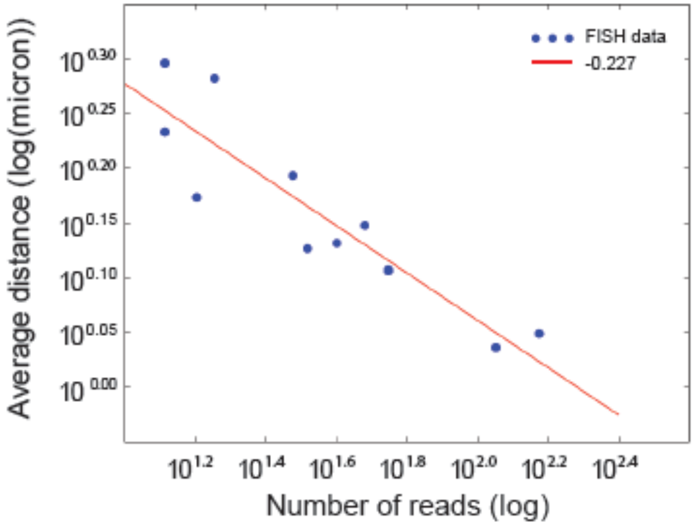


3D chromosomal
structure

Contact Frequency vs. Spatial Distance

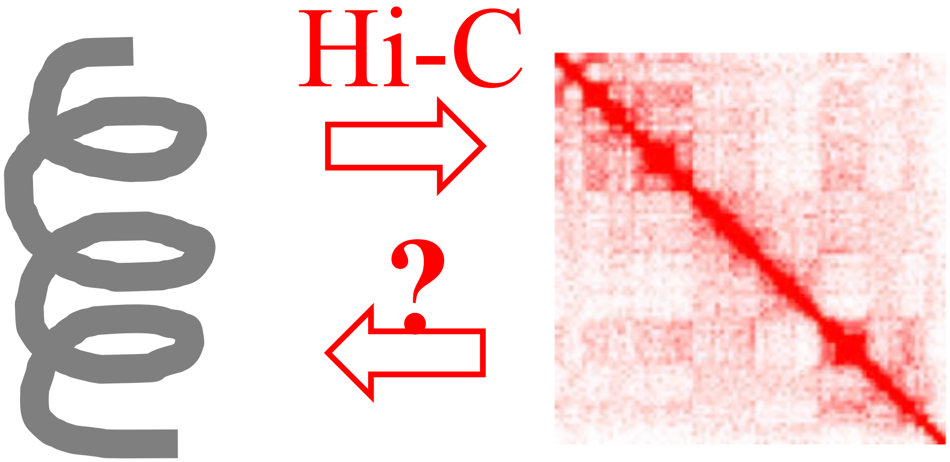






Lieberman-Aiden, et al, 2009

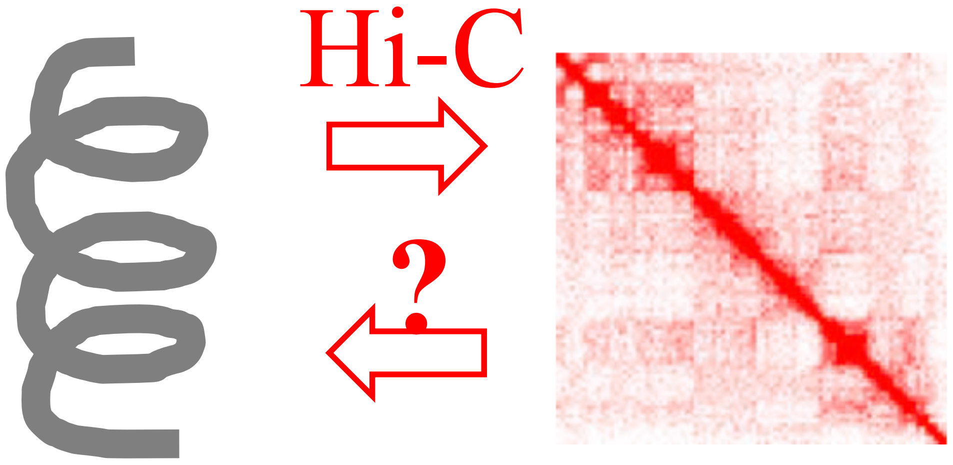
Problem setting



Hi-C

?

Problem setting

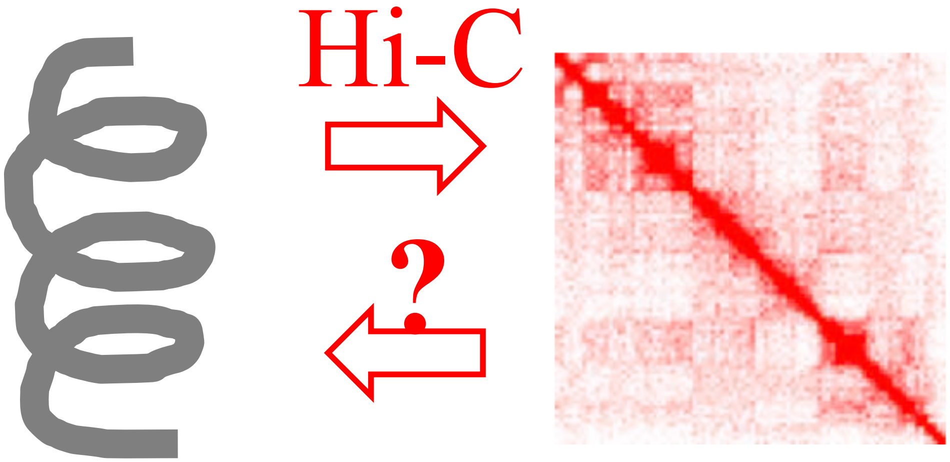


- Challenges:

➢ Sequencing uncertainties

➢ Biases: enzyme, GC content, mappability

Problem setting



- Challenges:

➢ Sequencing uncertainties

➢ Biases: enzyme, GC content, mappability

Yaffe and Tanay, 2011

Beads-on-a-string Representation

ACGTAGCTAGATACTGTAGTGTAGTTTGGAACCTGAGGG

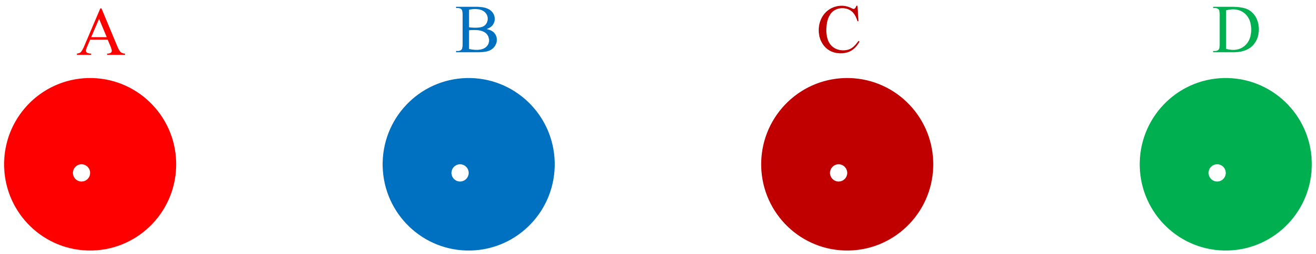
Beads-on-a-string Representation

ACGTAGCTAGATACTGTAGTGTAGTTTGGAACCTGAGGG

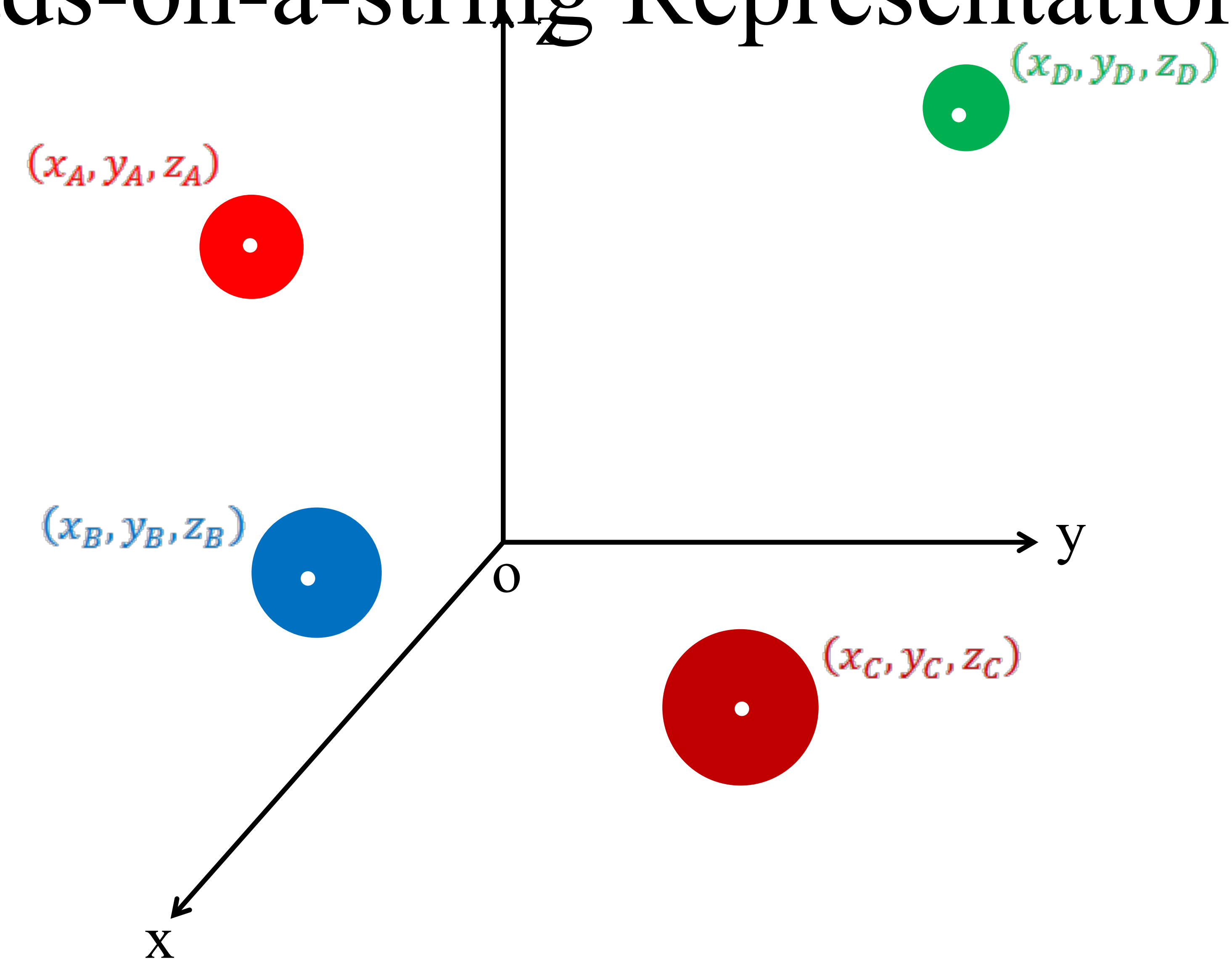
Beads-on-a-string Representation

ACGTAGCTAG ATACTGTAGT GTAGTTTGGA ACCTGAGGG

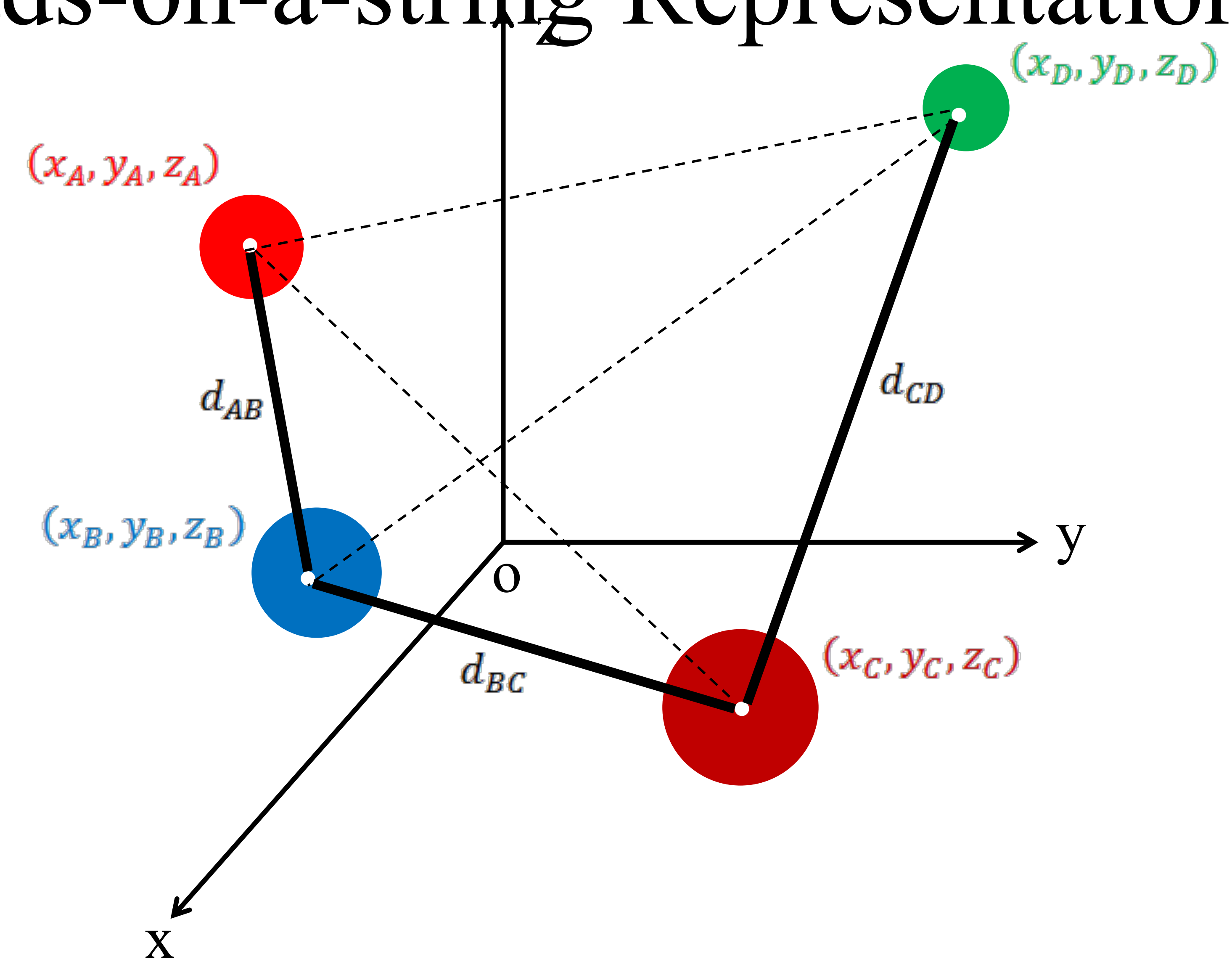
Beads-on-a-string Representation



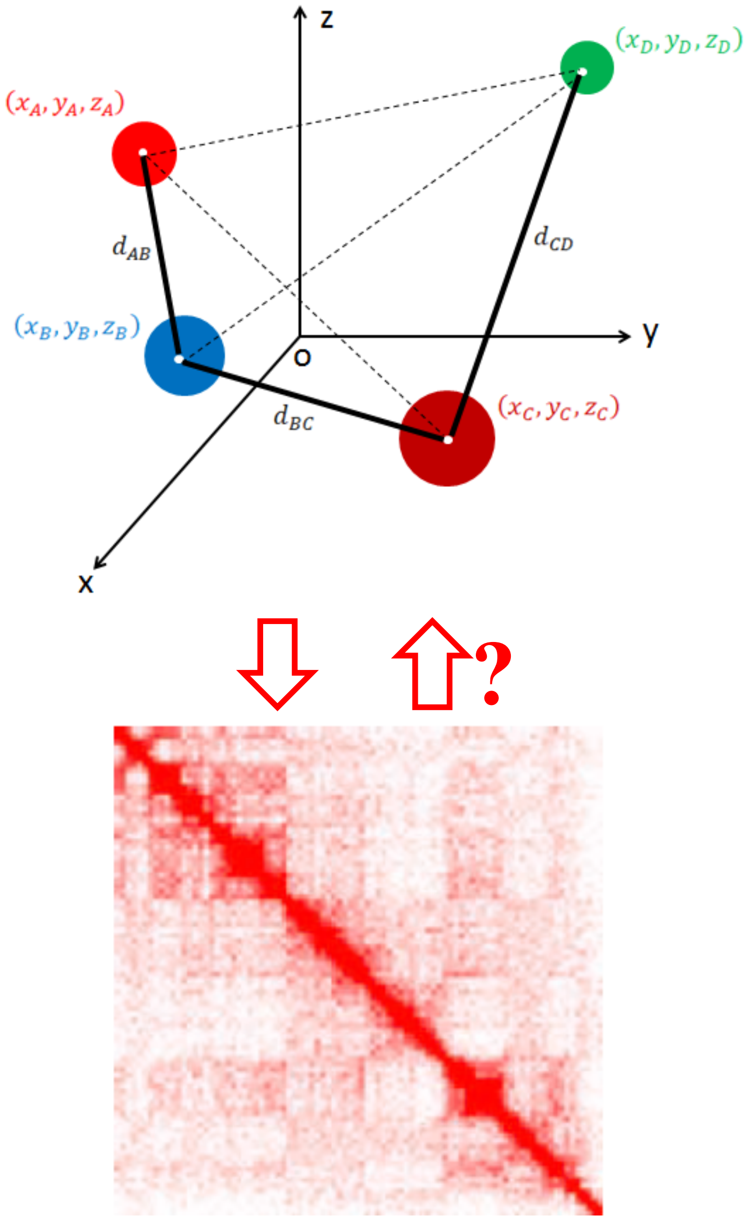
Beads-on-a-string Representation



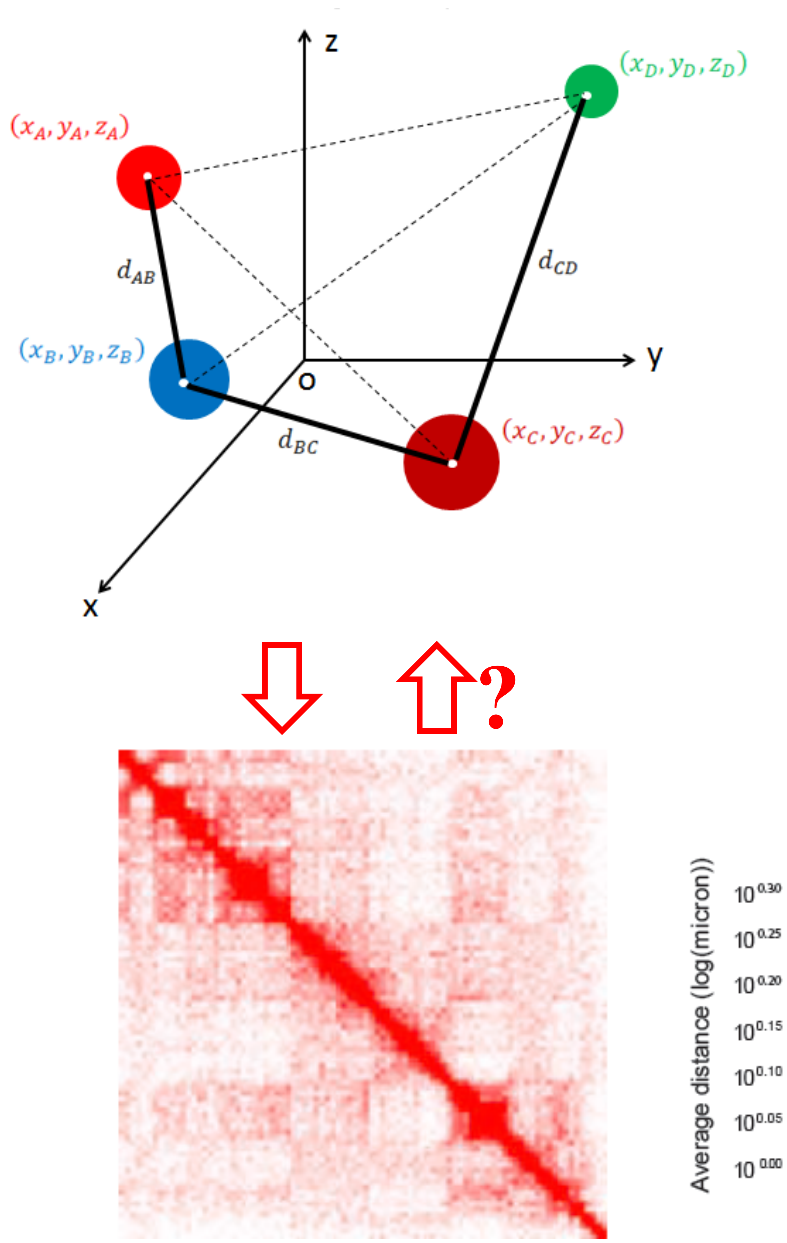
Beads-on-a-string Representation



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

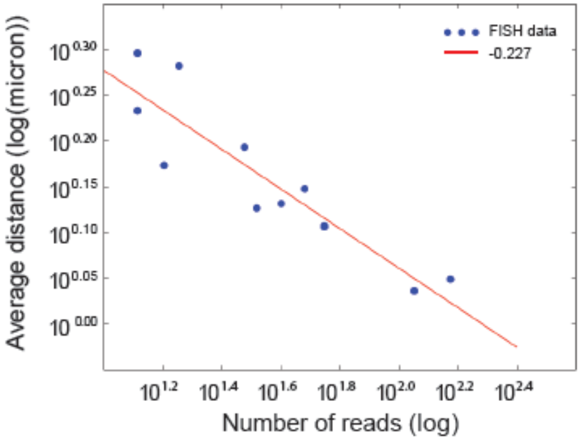
g_i : GC content of locus i

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:

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

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

Bayesian Statistical Model

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

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

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

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

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

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

Statistical Inference

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

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

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 $\beta_1 = -1$.

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

Statistical Inference

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

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

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

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

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

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 $\beta_1 = -1$.

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

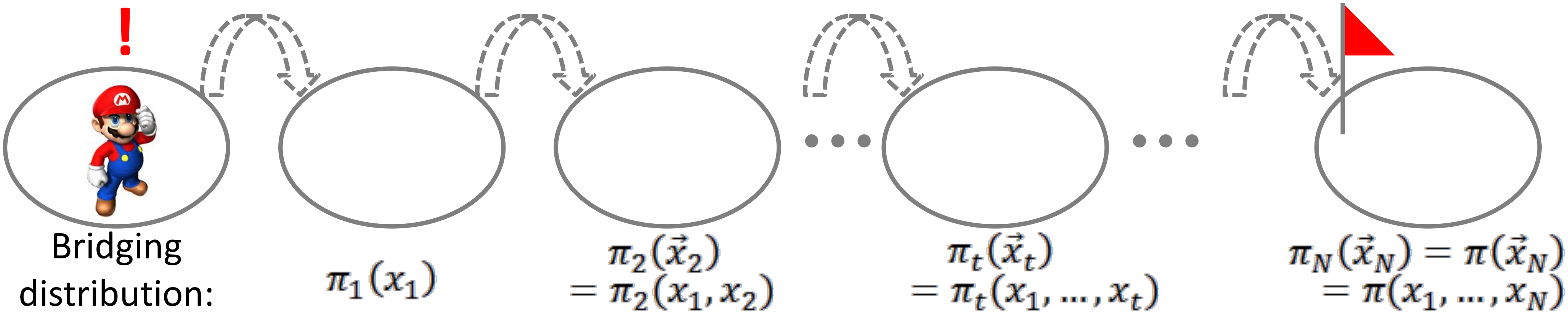
- Initialization 2: use sequential important sampling to get the initial 3D chromosomal structure $\{x_i, y_i, z_i, 1 \le i \le N\}$.

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

Sequential Importance Sampling

$$\pi(\vec{x}_N) = \pi(x_1, \ldots, x_N)$$

Sequential Importance Sampling



Bridging distribution:

$\pi_1(x_1)$

$\pi_2(\vec{x}_2)$
$= \pi_2(x_1, x_2)$

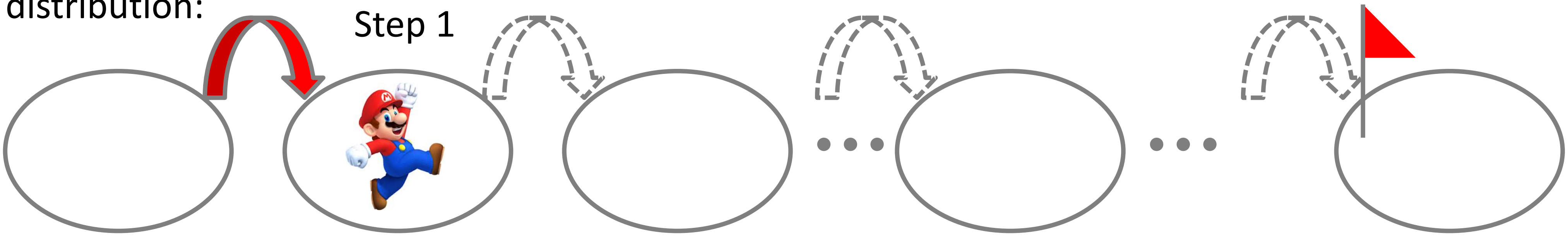
$\pi_t(\vec{x}_t)$
$= \pi_t(x_1, \dots, x_t)$

$\pi_N(\vec{x}_N) = \pi(\vec{x}_N)$
$= \pi(x_1, \dots, x_N)$

Sequential Importance Sampling

Proposal distribution: $g_1(x_1)$

Step 1



Bridging distribution: $\pi_1(x_1)$

$\pi_2(\vec{x}_2)$
$= \pi_2(x_1, x_2)$

$\pi_t(\vec{x}_t)$
$= \pi_t(x_1, \dots, x_t)$

$\pi_N(\vec{x}_N) = \pi(\vec{x}_N)$
$= \pi(x_1, \dots, x_N)$

Weighted samples:

$x_1 \sim g_1(x_1)$

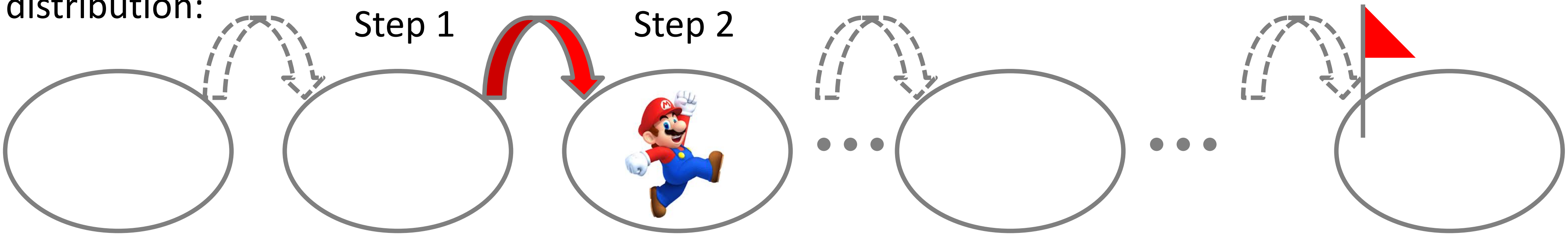
$$w_1 = \frac{\pi_1(x_1)}{g_1(x_1)}$$

(x_1, w_1)

Sequential Importance Sampling

Proposal distribution: $g_1(x_1)$ $g_2(x_2|x_1)$

Step 1 Step 2



Bridging distribution: $\pi_1(x_1)$

$$\pi_2(\vec{x}_2) = \pi_2(x_1, x_2)$$

$$\pi_t(\vec{x}_t) = \pi_t(x_1, \dots, x_t)$$

$$\pi_N(\vec{x}_N) = \pi(\vec{x}_N) = \pi(x_1, \dots, x_N)$$

Weighted samples: (x_1, w_1)

$$x_2 \sim g_2(x_2|x_1)$$

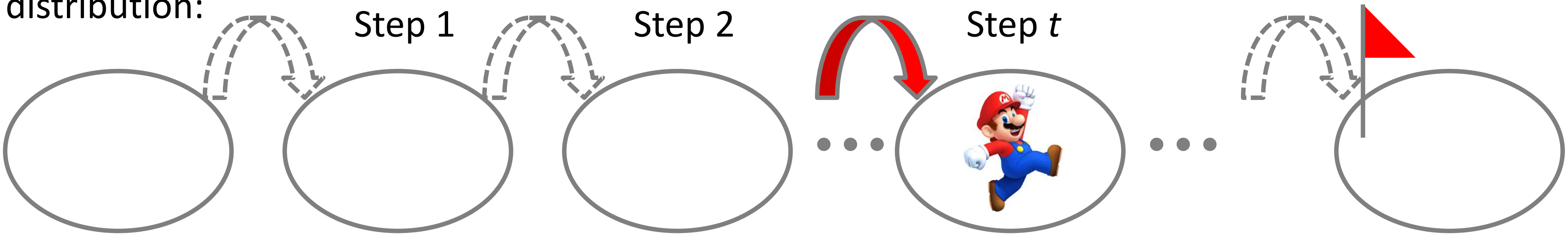
$$w_2 = \frac{w_1 \pi_2(\vec{x}_2)}{\pi_1(x_1) g_2(x_2|x_1)}$$

$$(\vec{x}_2, w_2)$$

Sequential Importance Sampling

Proposal distribution: $g_1(x_1)$ $g_2(x_2|x_1)$ $g_t(x_t|\vec{x}_{t-1})$

Step 1 Step 2 Step *t*



Bridging distribution:

$\pi_1(x_1)$

$\pi_2(\vec{x}_2) = \pi_2(x_1, x_2)$

$\pi_t(\vec{x}_t) = \pi_t(x_1, \dots, x_t)$

$\pi_N(\vec{x}_N) = \pi(\vec{x}_N) = \pi(x_1, \dots, x_N)$

Weighted samples:

(x_1, w_1)

$(\vec{x}_2, w_2)$

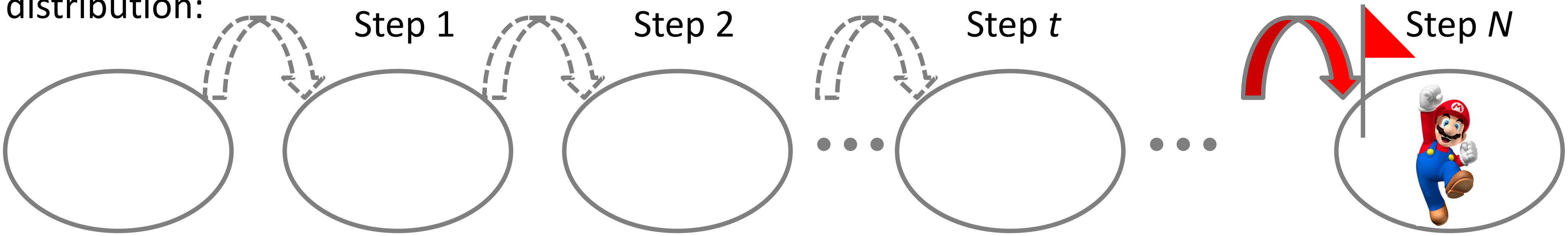
$x_t \sim g_t(x_t|\vec{x}_{t-1})$

$$w_t = \frac{w_{t-1}\pi_t(\vec{x}_t)}{\pi_{t-1}(\vec{x}_{t-1})g_t(x_t|\vec{x}_{t-1})}$$

$(\vec{x}_t, w_t)$

Sequential Importance Sampling

Proposal distribution: $g_1(x_1)$ $g_2(x_2|x_1)$ $g_t(x_t|\vec{x}_{t-1})$ $g_N(x_N|\vec{x}_{N-1})$



Bridging distribution:

$\pi_1(x_1)$

$\pi_2(\vec{x}_2) = \pi_2(x_1, x_2)$

$\pi_t(\vec{x}_t) = \pi_t(x_1, \dots, x_t)$

$\pi_N(\vec{x}_N) = \pi(\vec{x}_N) = \pi(x_1, \dots, x_N)$

Weighted samples:

(x_1, w_1)

$(\vec{x}_2, w_2)$

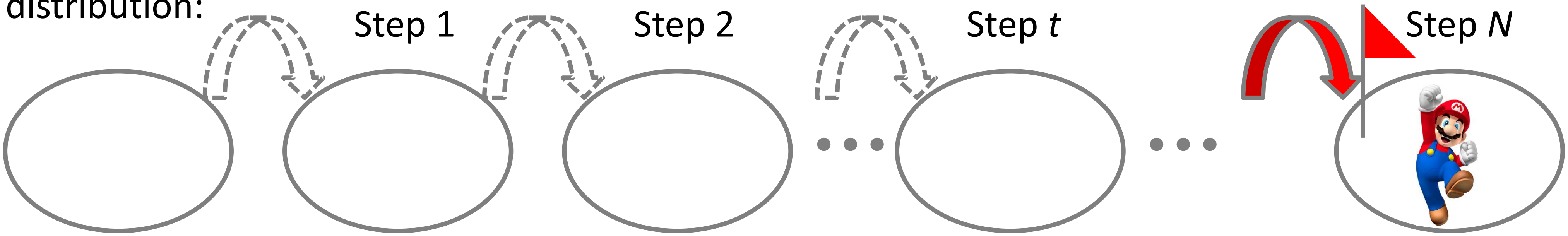
$(\vec{x}_t, w_t)$

$x_N \sim g_N(x_N|\vec{x}_{N-1})$

$$w_N = \frac{w_{N-1}\pi_N(\vec{x}_N)}{\pi_{N-1}(\vec{x}_{N-1})g_N(x_N|\vec{x}_{N-1})}$$

$(\vec{x}_N, w_N)$

Sequential Importance Sampling



Proposal distribution: $g_1(x_1)$ $g_2(x_2|x_1)$ $g_t(x_t|\vec{x}_{t-1})$ $g_N(x_N|\vec{x}_{N-1})$

Step 1 Step 2 Step t Step N

Bridging distribution: $\pi_1(x_1)$ $\pi_2(\vec{x}_2) = \pi_2(x_1, x_2)$ $\pi_t(\vec{x}_t) = \pi_t(x_1, \dots, x_t)$ $\pi_N(\vec{x}_N) = \pi(\vec{x}_N) = \pi(x_1, \dots, x_N)$

Weighted samples: (x_1, w_1) $(\vec{x}_2, w_2)$ $(\vec{x}_t, w_t)$ $(\vec{x}_N, w_N)$

Sequential Importance Sampling (SIS) Algorithm:

(1) Design bridging distributions $\pi_t(\vec{x}_t)$ and proposal distributions $g_t(x_t|\vec{x}_{t-1})$

(2) Sequentially draw weighted samples $x_t \sim g_t(x_t|\vec{x}_{t-1})$, and update weight

$$w_t = \frac{w_{t-1}\pi_t(\vec{x}_t)}{\pi_{t-1}(\vec{x}_{t-1})g_t(x_t|\vec{x}_{t-1})}$$

SIS in BACH: Outline

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

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

SIS in BACH: Outline

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

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

- Bridging distributions:

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

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:

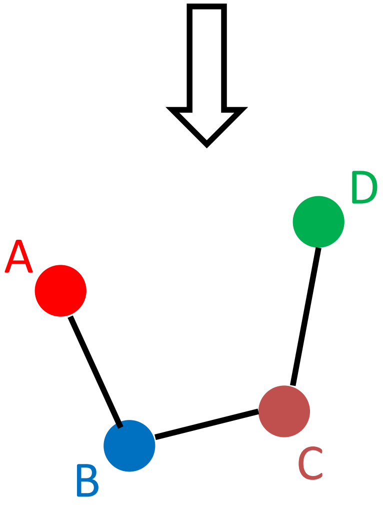
$$\pi_t(x_i, y_i, z_i, 1 \le i \le t | u_{ij}, 1 \le i < j \le t)$$

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

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

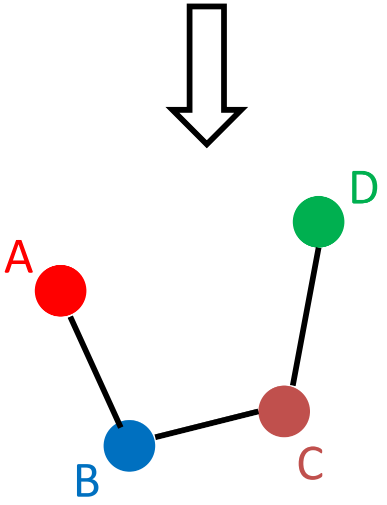
SIS in BACH: Illustration

	A	B	C	D
A		▨	▨	▨
B	▨		▨	▨
C	▨	▨		▨
D	▨	▨	▨	



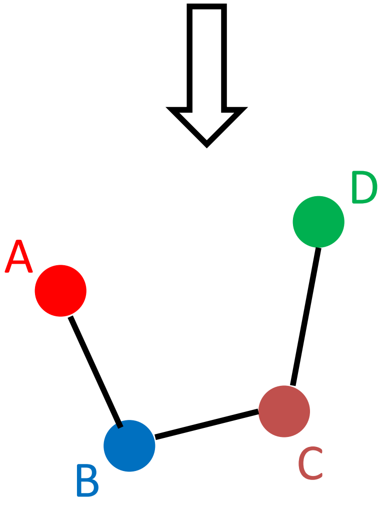
SIS in BACH: Illustration

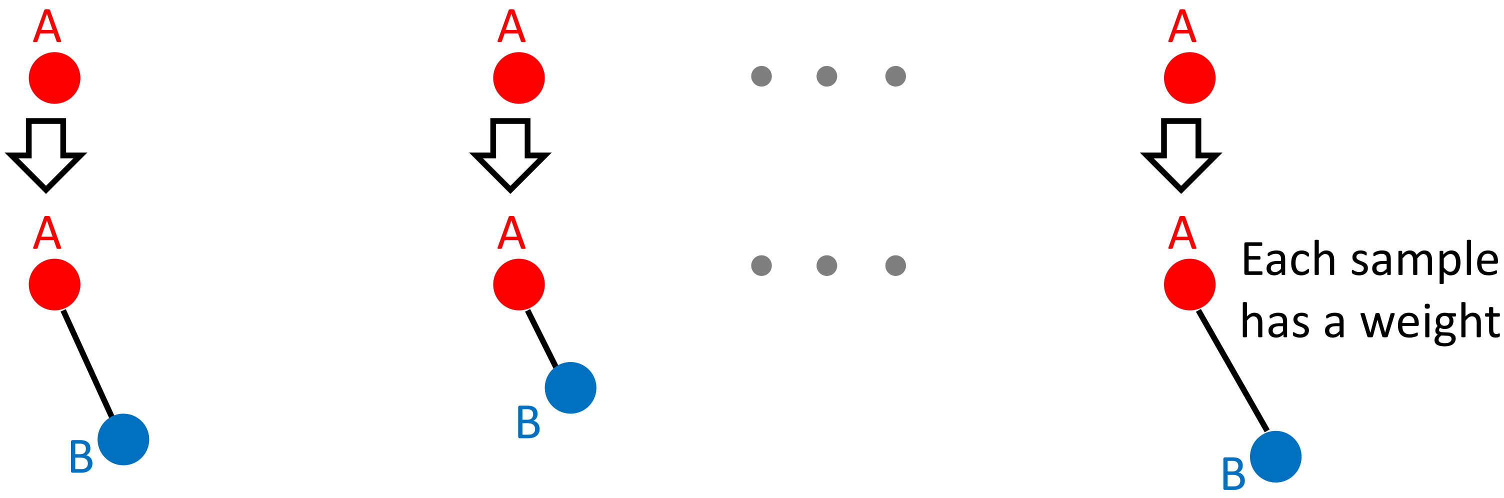
	A	B	C	D
A		▨	▨	▨
B	▨		▨	▨
C	▨	▨		▨
D	▨	▨	▨	



SIS in BACH: Illustration

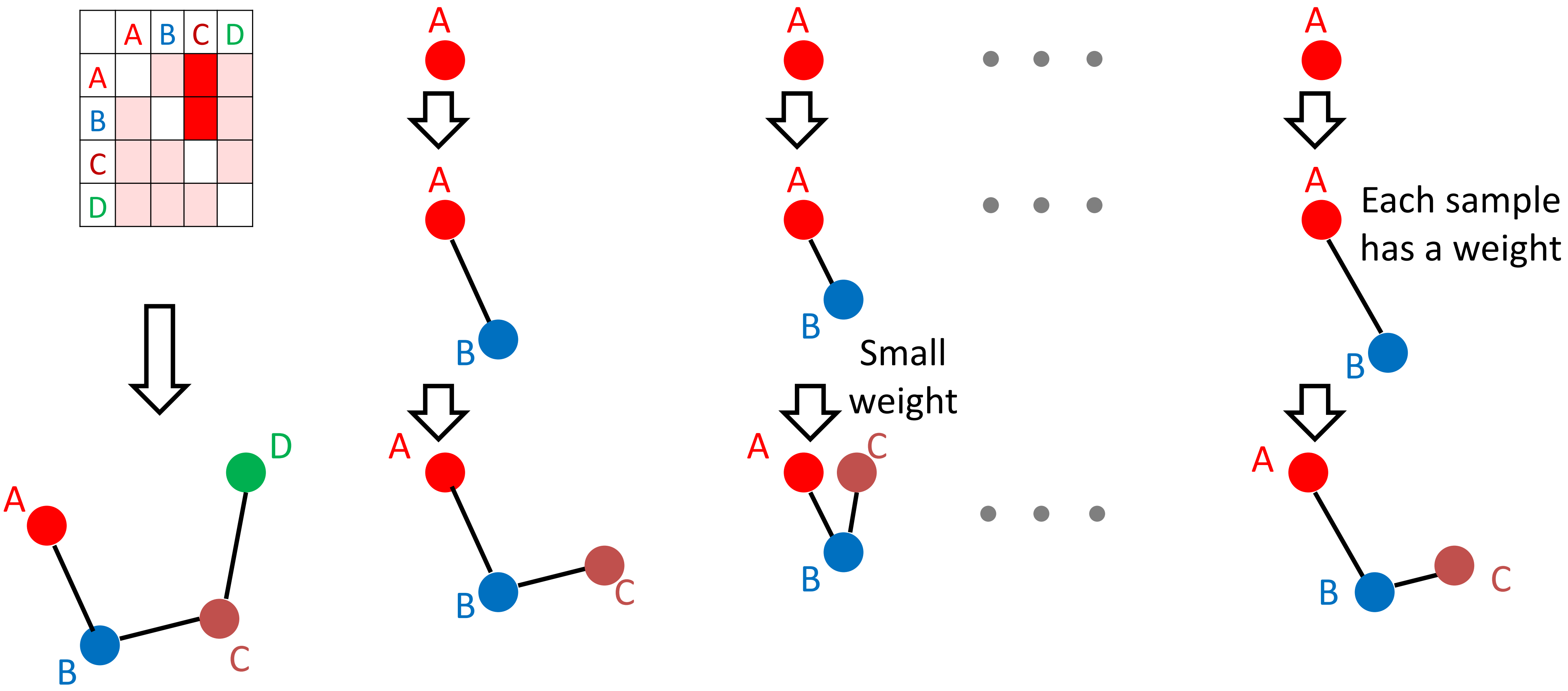
	A	B	C	D
A		🟥	▨	▨
B	▨		▨	▨
C	▨	▨		▨
D	▨	▨	▨	





Each sample has a weight

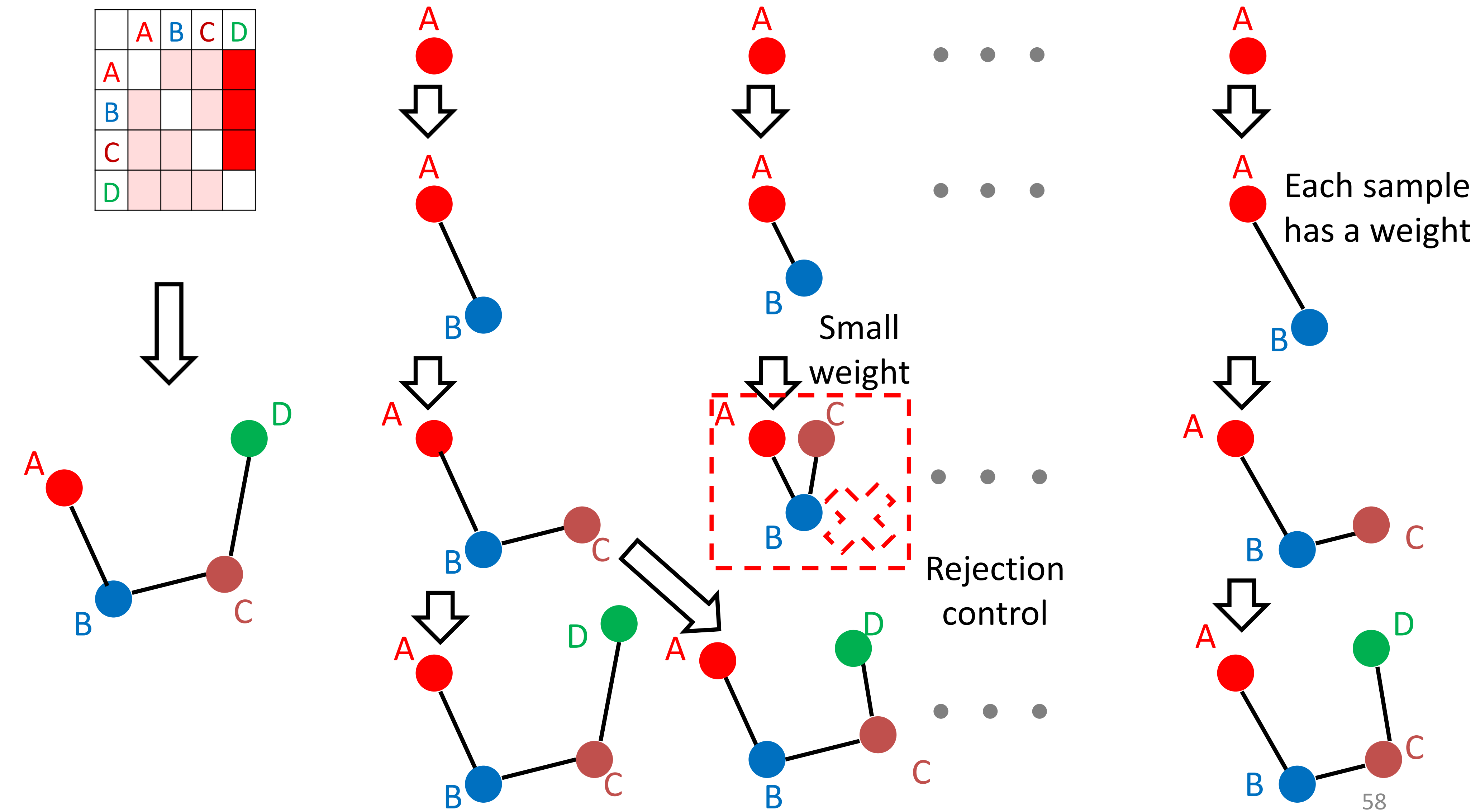
SIS in BACH: Illustration



Small weight

Each sample has a weight

SIS in BACH: Illustration



	A	B	C	D
A		▨	▨	🟥
B	▨		▨	🟥
C	▨	▨		🟥
D	▨	▨	▨	

Small weight

Rejection control

Each sample has a weight

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.

Duane, et al, 1987

Hybrid Monte Carlo in BACH

- Goal: sampling from

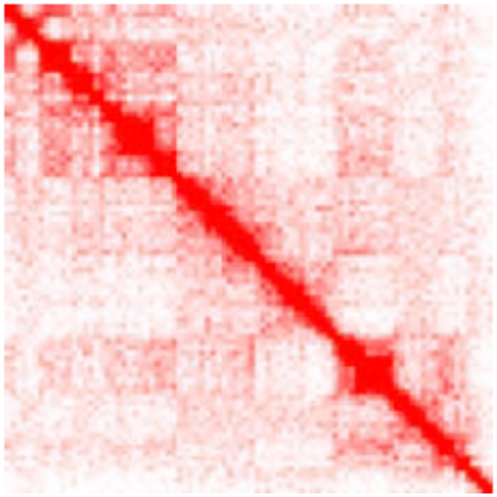
$$\pi(x_i, y_i, z_i, 1 \le i \le N | u_{ij}, 1 \le i < j \le N)$$

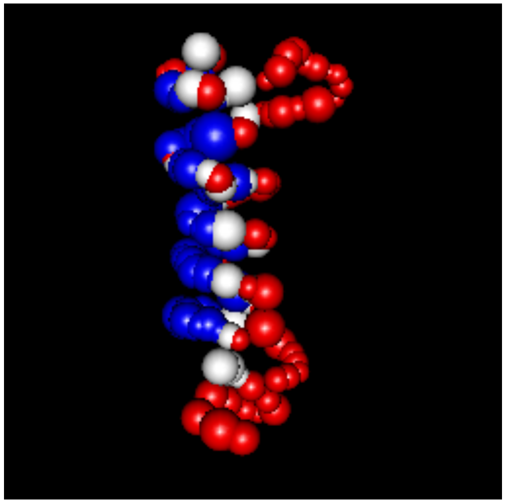
- Take partial derivate of log likelihood over 3D coordinates $(x_i, y_i, z_i, 1 \le i \le N)$.

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

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.





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.

Acknowledgements

Jun S. Liu

Ke Deng

Bing Ren

Jesse Dixon

Siddarth Selvaraj