

Supplementary Material

Automated mapping of large-scale chromatin structure in ENCODE

Probabilistic Models and Estimation Methods

0.1 Generative probability model

In the usual HMM approach, observations from a given state are modeled as being drawn from a specific distribution, commonly a normal distribution. As shown in the main text, we found that using this simple approach does not fit the data well. To address this poor fit, we designed a hierarchical model that uses a continuous mixture of normals to model each state; that is, we put a prior on the mean and variance of the observations given the state. Within each state, there can be different means and variances for different parts of the data. Estimation of the parameters of the CPM is only somewhat more demanding than for a simple HMM, since after analytically integrating out the hidden variables, we only need to find four parameters per state rather than the two per state required by the HMM. The time complexity of our recursion applied to each ungapped data segment is $O(n^2)$, while that of an HMM is $O(n)$ where n is the length of the ungapped data segment. As pointed out above, the time complexity for the full data set is $O(N)$ where N is the number of ungapped data fragments in the data set.

The CPM model can be described as follows. Let us assume for now that we know the distribution on the length of each segment, for each of the four states, and we know the transition probabilities between the states. Suppose the maximum number of change points is k_{max} . Denote a segmentation by $A = \{\delta_0, c_1, \delta_1, c_2, \dots, \delta_{k-1}, c_k, \delta_k\}$, $1 \leq c_1 < c_2 < \dots < c_k \leq n$, c_i is the i th change point, and δ_i is the state between its neighboring change points taking value in the set $S = \{0, 1, 2, 3\}$ (by convention, the state at the change point c_i is δ_{i-1}). The prior probability of A before observing the data is

$$p(A) = \prod_{i=0}^k p_{\delta_i}(c_{i+1} - c_i + 1) \cdot \pi(\delta_0) \prod_{i=0}^{k-1} K(\delta_i, \delta_{i+1}), \quad (1)$$

where $p_{\delta}(l)$ is the probability of the length of one ungapped data segment being l , given the segment state is δ , and K is the transition probability between states with initial distribution

π . Note that we put transition probabilities between segments, instead of between each data point as in the traditional HMM. Also, transitions between the same state are allowed. Larger units consisting of one or more adjacent segments with the same state are called “fragments.” We set $c_0 = 0$ and c_{k+1} to be the last data point.

Given the segmentation A , the mean and variance for each segment is generated from a normal-inverse- χ^2 distribution:

$$\mu_i | \sigma_i^2, A \sim N\left(\mu_{\delta_i}^{(h)}, \frac{\sigma_i^2}{k_{\delta_i}^{(h)}}\right) \quad (2)$$

$$\sigma_i^2 | A \sim Inv - \chi^2(\nu_{\delta_i}^{(h)}, \sigma_{\delta_i}^{2(h)}) \quad (3)$$

Those parameters with superscript (h) are hyperparameters that need to be specified.

With segmentation A and μ_i, σ_i given for each segment, the observations are naturally modeled as normal with given mean and variance:

$$y_{c_i+1:c_{i+1}} | \mu_i, \sigma_i^2, A \stackrel{iid}{\sim} N(\mu_i, \sigma_i^2) \quad (4)$$

where we use the notation $y_{i:j} = \{y_k | k = i, i + 1, \dots, j\}$

0.2 Computing the posterior distribution

We model the segment length using the $Gamma(\alpha, \beta)$ distribution. In this case, the hyperparameters are $\Theta = \{\mu_{\delta}^{(h)}, k_{\delta}^{(h)}, \nu_{\delta}^{(h)}, \sigma_{\delta}^{2(h)}, K(\delta, \delta'), \pi(\delta), \alpha_{\delta}, \beta_{\delta} : \delta, \delta' \in \{0, 1, 2, \dots\}\}$. For now, if we assume that these parameters are known, then the main goal of inference is to get at the posterior distribution $P(A|y) = p(A)p(y|A) / \sum_{A'} p(A')p(y|A')$. The direct summation over all different segmentations is computationally prohibitive since the number of possible segmentations increases with n like $n^{k_{max}}$. A dynamic programming recursion similar to the one used in [4] can be used to reduce the complexity to $O(n^2k)$ as follows.

Let $p(i, \delta, k)$ be the probability that the first k segments have a total length i and the last segment is in state δ . These probabilities can be computed using the following recursion:

$$p(i, \delta, 1) = \pi_{\delta} p_{\delta}(i) \quad (5)$$

$$p(i, \delta, k) = \sum_{j < i, \delta'} p(j, \delta', k-1) p_{\delta}(i-j) K(\delta', \delta) \quad (6)$$

$$(7)$$

The main recursion can be computed as follows:

$$Pr(y_{1:i} | y_{1:i} \text{ has } k \text{ segments } (k-1 \text{ change points}), \text{ with last state } \delta) \quad (8)$$

$$= \sum_{j < i, \delta'} Pr(y_{1:i}, \text{ previous change point is at } j, \text{ the state ending at } j \text{ is } \delta' | \delta, k) \quad (9)$$

$$= \sum_{j < i, \delta'} P(j, \delta', k - 1 | i, \delta, k) P(y_{1:j} | \delta', k - 1) P(y_{j+1:i} | \delta, 1) \quad (10)$$

$$= \sum_{j < i, \delta'} \frac{p(j, \delta', k - 1) p_\delta(i - j) K(\delta', \delta)}{p(i, \delta, k)} P(y_{1:j} | \delta', k - 1) P(y_{j+1:i} | \delta, 1) \quad (11)$$

$$(12)$$

All of the probabilities above should be understood as being conditioned on the hyperparameters, which we omit for simplicity in notation.

The base case when $k = 1$ for the above recursion can be obtained by integrating out the hidden variables:

$$P(y_{i:j} | \delta, 1) = \int P(y_{i:j} | \mu, \sigma^2) p(\mu, \sigma^2 | \mu_\delta^{(h)}, k_\delta^{(h)}, \nu_\delta^{(h)}, \sigma_\delta^{2(h)}) d\mu d\sigma^2 \quad (13)$$

The integration can be done analytically since we used the usual conjugate prior in the model.

After the recursion, all of the desired probabilities can be easily obtained. For example, the posterior probability of the number of segments can be obtained by first computing

$$p(k, \text{ with last state } \delta | y_{1:n}) \propto p(n, \delta, k) p(y_{1:n} | \delta, k) \quad (14)$$

and then summing over δ . The marginal likelihood is just

$$p(y_{1:n}) = \sum_{k, \delta} p(n, \delta, k) p(y_{1:n} | \delta, k). \quad (15)$$

Using the marginal likelihood, we can also compute the probability of a specific segmentation by $p(A|y) = p(A)p(y|A)/p(y)$.

We can also sample from the posterior distribution on the segmentations given the data. First we draw a value for the number of segments together with the last state of segmentation from the posterior $p(k, \delta | y)$, which we have derived above. Then we can sample backward from the following distribution:

$$P(c_{m-1} = j, \text{ with state } \delta' | y_{1:n}, c_m = i, \text{ with state } \delta) \quad (16)$$

$$\propto p(y_{1:j} | \delta', k - 1) \cdot p(y_{j+1:i} | \delta, 1) p(j, \delta, k - 1) p_\delta(i - j) K(\delta', \delta) \quad (17)$$

If a single estimator is desired, the most frequently used one is the maximum *a posteriori* (MAP) estimate. Here we also use the ‘‘centroid’’ deduced from samples as our estimate [3]. In our case, the centroid can be obtained as follows. We first obtain n samples (500 in our implementation). In general the centroid is the admissible solution that is the minimum distance from the sampled solutions. Here a admissible solution is an assignment of a state to each probe, and the distance between to solutions is the number of probes state assignments

	ENm001	ENm002	ENm003	ENm004	ENm005	ENm006
CI of Centroid	0.087	0.123	0.083	0.098	0.102	0.076
CI of MAP	0.115	0.153	0.125	0.132	0.134	0.091
	ENm007	ENm008	ENm009	ENm010	ENm011	ENm012
CI of Centroid	0.145	0.149	0.092	0.088	0.187	0.118
CI of MAP	0.197	0.195	0.141	0.121	0.231	0.152
	ENm013	ENm014	ENr111	ENr112	ENr113	ENr114
CI of Centroid	0.083	0.114	0.082	0.078	0.118	0.140
CI of MAP	0.141	0.139	0.139	0.093	0.155	0.184
	ENr121	ENr122	ENr123	ENr131	ENr132	ENr133
CI of Centroid	0.139	0.147	0.143	0.104	0.095	0.086
CI of MAP	0.194	0.183	0.175	0.141	0.143	0.127
	ENr211	ENr212	ENr213	ENr221	ENr222	ENr223
CI of Centroid	0.091	0.139	0.086	0.089	0.134	0.096
CI of MAP	0.110	0.182	0.122	0.132	0.169	0.122
	ENr231	ENr232	ENr233	ENr311	ENr312	ENr313
CI of Centroid	0.107	0.145	0.088	0.099	0.112	0.134
CI of MAP	0.142	0.185	0.134	0.133	0.149	0.173
	ENr321	ENr322	ENr323	ENr324	ENr331	ENr332
CI of Centroid	0.136	0.138	0.126	0.134	0.084	0.092
CI of MAP	0.157	0.185	0.141	0.159	0.128	0.148
	ENr333	ENr334				
CI of Centroid	0.130	0.122				
CI of MAP	0.199	0.146				

Table 1: Credibility intervals for the centroid estimate and the MAP.

that do not agree. In this case the centroid is the set of probe assignments that correspond to the state most frequently observed in the sample for each probe. An advantage of using samples is that we immediately get a sense of the uncertainty from the estimated probability of being in the centroid state, based on the percentage of all those n samples in the centroid state.

As an overall measure of the quality of a solution we employ a 90% credibility limit, where the 90% credibility limit is the distance from the solution that includes 90% of the sample solutions. Since our samples are drawn directly from the posterior distribution, these limits are sampling estimates of the Bayesian posterior credibility limits [1]. Table 1 gives the 90% credibility limits for the MAP solution and the centroid solution. For example for ENCODE regions ENm001 we find that 90% of the sampled solutions have no more than 8.7% probes whose state is assigned differently from that of the centroid solution. Although the credibility limits for both solutions are generally tight we see that the centroid limits are generally 20 to 30% narrower than the MAP credibility limits.

Finally, posterior mean and variance at each point can be estimated by first drawing multiple segmentations from the posterior $p(A|y)$. Given a segmentation, the posterior mean for the hidden parameters, the mean μ_i and the variance σ_i^2 can be obtained from $p(\mu_i, \sigma_i^2 | \delta_i, y)$ since we used a conjugate prior. The estimate for posterior mean and variance at each point is obtained after averaging over multiple segmentations. Under the assumptions of our model, the residual standardized by the posterior estimate of mean and variance at each point should be normally distributed. From the plot in the main text, we can see that for

both models, the residuals for states 1 and 2 are reasonably normally distributed, but for states 0 and 3 the residuals of the HMM depart strongly from a normal distribution.

0.3 Estimating model parameters

We have, up to now, assumed that the model parameters are fixed in advance. In practice, setting those parameters is not a simple task. In this study, use an empirical Bayes approach; that is, we choose the parameters that maximize the likelihood $p(y_{1:n}|\Theta)$.

Directly maximizing this likelihood is impractical for two reasons. First, for each fixed set of parameters, evaluating the likelihood would require doing the above recursion from the start. The optimization procedure would require such evaluations many times, which is infeasible. Second, optimization over parameters of more than 20 dimensions is difficult even with state-of-the-art optimization techniques. A natural choice to avoid this difficulty is the expectation-maximization (EM) algorithm [2], considering the segmentation as the missing variable. In the E-step, we need to compute

$$\sum_A p(A|y, \Theta^{(old)}) \log p(A, y|\Theta). \quad (18)$$

Unfortunately, summing over all possible segmentations is computationally intensive, even when using the dynamic programming recursion, because many such evaluations are required for EM. To avoid this summation, we use stochastic EM, in which samples from the distribution $p(A|y, \Theta^{(old)})$ are used to approximate the summation in 18. These samples are drawn from the posterior, using parameters from the previous iteration:

$$E[\log P(A, y|\Theta)|y, \Theta^{(old)}] \approx \frac{1}{N} \sum_{n=1}^N \log p(A^{(n)}, y|\Theta), \quad (19)$$

where $\{A^{(n)}\}_1^N$ are samples from $p(A|y, \Theta^{(old)})$.

The sum $\sum \log p(A^{(n)}, y|\Theta)$ can be written as

$$\begin{aligned} \sum \log p(A^{(n)}, y|\Theta) &= \sum \log p(A^{(n)}|\Theta)p(y|A^{(n)}, \Theta) \\ &= \log \prod_{\delta, \delta'=1}^D K(\delta, \delta')^{n_{\delta\delta'}} + \log \prod_{\delta=1}^D \pi(\delta)^{n_\delta} + \\ &\quad \sum_{\delta} \sum_{(i,j) \in A_\delta} \log p_\delta(i-j+1|\alpha_\delta, \beta_\delta) + \\ &\quad \sum_{\delta} \sum_{(i,j) \in A_\delta} \log P(y_{i:j}|\delta, 1, \Theta_\delta) \end{aligned}$$

where $n_{\delta\delta'}$ is the count of the number of transitions from δ to δ' in $\{A^{(n)}\}_1^N$. n_δ is the number of times $\{A^{(n)}\}_1^N$ starts in state δ , A_δ contains all segments of $\{A^{(n)}\}_1^N$ that is in state δ , and

$\Theta_\delta = \{\mu_\delta^{(h)}, k_\delta^{(h)}, \nu_\delta^{(h)}, \sigma_\delta^{2(h)}\}$ is the part of hyperparameters in the conjugate prior related to state δ in the observation model.

One important observation is that in the above, the optimization can be carried out independently for each summand. The hyperparameters Θ has been partitioned into four parts. The first part consists of those in the transition matrix for the state transition probability. The second part consists of initial probabilities for each of the states. The third part consists of the shape and scale parameters of the Gamma distribution used in modeling the length of each segment. Last, the hyperparameters for the priors on the hidden parameters, the mean and variance of each segment, are separated from other hyperparameters. Moreover, the parameters for different states are separated. This is the result of conditioning on the sampled segmentations, which decouples the hyperparameters into different groups, which makes our optimization problem much easier. Actually we only have optimization problems with no more than 4 dimensions. The optimized value for parameters $K(\delta, \delta')$ and $\pi(\delta)$ is easily seen to be $n_{\delta, \delta'} / \sum_{\delta'} n_{\delta, \delta'}$ and $n_\delta / \sum_{\delta'} n_{\delta, \delta'}$ respectively, and the other parameters are optimized using IMSL function *min_col_gen_lin*. A simpler way to estimate the shape and the scale parameters are as follows. For each state δ , we collect all the segments from the samples that is in state δ , then we compute the mean m_δ and variance v_δ of the lengths. The shape parameter α_δ and scale parameter β_δ is estimated by matching the first two moments:

$$\alpha_\delta / \beta_\delta = m_\delta, \quad \alpha_\delta / \beta_\delta^2 = v_\delta$$

, which leads to

$$\alpha_\delta = m_\delta^2 / v_\delta, \quad \beta_\delta = m_\delta / v_\delta$$

. This is simpler than doing numerical optimization, although it does not fit within our framework of maximizing the likelihood.

The recursion still needs to be done multiple times, once for each iteration. In practice, convergence is observed within less than 10 iterations. For this training, we employed a subset of the available data: all ungapped sequences with at least 100 probes.

0.4 Handling gaps

Our model also yields a principled means to span gaps in the data that goes beyond the implicit geometric length distributions of HMMs. When the gaps are small, we expect the states of the bases at the corresponding edges to be similar, but this correlation across gaps is unlikely to span long gaps. Our models describing the lengths of substrings and state transitions yield inferences on the state at the probe just adjacent to a gap as a function of gap length. The inference of this missing data borrows information from the other side of the gap. Over a gap, our gamma distribution model of substring length infers the number of change points in the gap, and the state transition model determines the distribution of states after k state transitions. This characteristic facilitates scaling up, since the basic recursion,

which is $O(n^2)$ operations, comes to bear over the data fragments rather than over the much longer full length regions spanned by the tiled array.

0.5 Enrichment calculation

The enrichment calculation assumes a uniform distribution of annotated elements as a null model. Thus if state 0 covers B_0 bases, and state 1 covers B_1 bases, then the expected fraction of the total number of annotated elements from state 0 and state 1 that fall in state 0 is $E_0 = B_0/(B_0 + B_1)$, and the enrichment/depletion of the element in state 0 is $(X_0 - E_0)/E_0$, where X_0 is the observed fraction of elements (from the total in state 0 and state 1) that fall in state 0. Similarly for state 1. Relative enrichment of state 1 with respect to state 0 is defined as the difference of the two.

References

- [1] Carlin and Thomas. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, London, 1996.
- [2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–22, 1977.
- [3] Y. Ding, C. Chan, and C. E. Lawrence. Clustering of RNA secondary structures with applications to messenger RNAs. *RNA*, 11:1157–1166, 2005.
- [4] J. S. Liu, F. Neuwald, and C. E. Lawrence. Bayesian models for multiple local sequence alignment and gibbs sampling strategies. *Journal of the American Statistical Association*, 90(432), December 1995.