

# CHiCAGO: Robust Detection of DNA Looping Interactions in Capture Hi-C data

Additional file 1: The mathematical specification of the  
CHiCAGO algorithm

Jonathan Cairns\*, Paula Freire-Pritchett\*, Steven W. Wingett,  
Andrew Dimond, Vincent Plagnol, Daniel Zerbino,  
Stefan Schoenfelder, Biola-Maria Javierre, Cameron Osborne,  
Peter Fraser and Mikhail Spivakov

\* Joint lead authors

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Definitions</b>	<b>2</b>
<b>3</b>	<b>Model</b>	<b>3</b>
<b>4</b>	<b>Estimation – Brownian noise</b>	<b>4</b>
<b>5</b>	<b>Estimation – Technical noise</b>	<b>7</b>
<b>6</b>	<b>Calculating <math>p</math>-values</b>	<b>8</b>
<b>7</b>	<b>Working with multiple replicates</b>	<b>9</b>
<b>8</b>	<b>Multiple testing and <math>p</math>-value weighting</b>	<b>9</b>
<b>A</b>	<b>Mean bin value</b>	<b>12</b>

# 1 Introduction

This document describes, in detail, the statistical framework that underpins the CHiCAGO algorithm.

## 2 Definitions

- The genome is partitioned into *restriction fragments*, according to our choice of restriction enzyme. We follow the HiC protocol, making pairs of fragments ligate together. Subsequently, the capture step enriches for fragments of interest.

A *bait* is a fragment that is captured by this last step. An *other end* is a fragment ligated to a bait. Thus, any fragment can be an other end, baits included.

- $i, j$  are indices that refer to restriction fragments.  $i$  and  $j$  index over other ends and baits, respectively. With  $I$  other ends and  $J$  baits, we have  $i \in \{1, \dots, I\}$  and  $j \in \{1, \dots, J\}$ . Since an other end can be a bait, we have that each baited fragment has both an  $i$  and  $j$  index.\*
- Let  $X_{ij}$  be the number of observed read pairs that span from other end  $i$  to bait  $j$ .
- Let a “pair” be some choice  $(i, j)$ .
- *cis* and *trans* are abbreviations for cis-chromosomal and trans-chromosomal, respectively.
- Let  $d_{ij}$  be the genomic distance between the midpoints of fragments  $i$  and  $j$  (thus,  $d_{ij} \geq 0$ ). If  $(i, j)$  is a trans pair, then we assume  $d_{ij}$  is infinite.
- Often we need to group things by genomic distance. Thus, we define genomic distance bins:

$$B_0 = [0, w), B_1 = [w, 2w), \dots, B_b = [bw, (b + 1)w)$$

---

\*Note: in the software package, the indices  $i$  and  $j$  are not used – rather, each fragment gets an ID according to its genomic location. The ID is referred to as a baitID or otherEndID depending on context. Thus, if we take a list of potential baitIDs, they need not be contiguous.

Let  $d_b$  be the midpoint of bin  $b$ . Thus, we can rewrite the definition of  $B_b$  as

$$B_b = [d_b - w/2, d_b + w/2)$$

- The Negative Binomial (NB) distribution will be parametrized throughout this document in terms of the mean  $\mu$  and the dispersion parameter  $r$  (also known as the size parameter). Thus, if  $X \sim NB(\mu, r)$ ,

$$Var(X) = \mu + \frac{\mu^2}{r}$$

### 3 Model

The aim of CHiCAGO is to find *interaction events*: pairs of loci that are brought together by some protein complex, in a manner that occurs more often than by chance were that complex not there. Under the null hypothesis (*ie* in the absence of such a complex), we assume that a count  $X_{ij}$  is a convolution of two elements:

$$X_{ij} = B_{ij} + T_{ij}$$

These two components are:

- a Brownian motion noise component with NB distribution,  $B_{ij} \sim NB(\mu_{ij}, r)$ . This count represents read pairs that arise from the random collisions due to Brownian motion of the chromosome. (Note that the term “Brownian motion noise” is not to be confused with the statistical concept of “Brownian noise”.) Thus, the mean  $\mu_{ij}$  decays with distance. We assume that  $\mu_{ij}$  is the product of a bait fragment-specific bias  $s_j$ , an other end fragment-specific bias  $s_i$ , and some “distance profile”  $f$  that depends on the distance between the fragments:

$$\mu_{ij} = s_i s_j f(d_{ij})$$

where

$$f(d) \rightarrow 0 \text{ as } d \rightarrow \infty$$

Additional constraints are required to make this model identifiable (for example, the alternative solutions  $s'_i = \alpha s_i$ ,  $s'_j = \frac{s_j}{\alpha}$  have the same associated  $\mu_{ij}$ ). Thus, we set:

$$\sum_i \log(s_i) = \sum_j \log(s_j) = 0 \quad (1)$$

As a result,  $f(d_{ij})$  represents the frequency of local interactions that an “average” bait would exhibit.

- a technical noise component,  $T_{ij} \sim Pois(\lambda_{ij})$ . This corresponds to reads introduced by assay artefacts, such as sequencing errors, and thus  $T_{ij}$  counts read pairs that did not arise from contact events. It is assumed that  $T_{ij}$  does not depend on distance. However, we permit complex non-multiplicative errors (see Section 5).

In practice, we find that  $T_{ij}$  is very small.

We estimate each of the parameters  $f$ ,  $s_j$ ,  $s_i$ ,  $r$  and  $\lambda_{ij}$  in turn.

## 4 Estimation – Brownian noise

For a pair where the bait and other end are close, the technical noise is negligible compared to the Brownian noise – that is,  $\mu_{ij} \gg \lambda_{ij}$ .

Thus, under the null model, we can assume that

$$X_{ij} \sim NB(\mu_{ij}, r)$$

where

$$\mu_{ij} = s_i s_j f(d_{ij})$$

Initially, we aim to find the quantities  $s_j$  and  $f(d)$ . This must be done in a way that is robust against true interactions in the data.

### Estimating $f(d)$

Note that, when estimating  $f(d)$ , we always ignore bait-to-bait pairs. That is, when summing over other ends  $i$ , we exclude all other ends  $i$  that are also baited fragments.

For a given bait  $j$ , we define genomic bins  $B_b$  that tile a region of 3mb centred at the bait fragment. For the HindIII restriction enzyme, we set the

bin width  $w = 20\text{kb}$  (approximately 5 restriction fragments). First, calculate the average count over all of the other ends in a given bin  $b$ :

$$\bar{X}_{bj} = \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} X_{ij}$$

where  $n_{bj}$  is the number of other ends in bin  $b$ .

We have that

$$\mathbb{E}(\bar{X}_{bj}) \approx s_j f(d_b)$$

(see Appendix, Section A.)

Any observations with  $\bar{X}_{bj} = 0$  are censored. This is to ensure that zeros do not cause numerical instabilities in the next step.

We estimate  $f(d_b)$  as the geometric mean count over all bins at distance  $d_b$  – that is,

$$\hat{f}(d_b) = \text{geo}_j \bar{X}_{bj}$$

equivalently,

$$\log \hat{f}(d_b) = \frac{1}{J} \sum_j \log(\bar{X}_{bj})$$

This method is similar to the size factor estimation procedure in DESeq (ANDERS and HUBER, 2010). We confirmed the accuracy with a simulation study (data not shown).

To get from  $\hat{f}(d_b)$  to full inference of the function  $\hat{f}(d)$ , we fit a cubic function on a log-log scale, extrapolating linearly beyond the given  $d_b$  values assuming continuity of  $f(d)$  and  $f'(d)$ .

## Estimating $s_j$

When estimating  $s_j$ , bait-to-bait pairs are ignored as for  $f(d)$  estimation. From the previous Section, we have

$$\mathbb{E}(\bar{X}_{bj}) \approx s_j f(d_b)$$

Thus, using our estimate  $\hat{f}(d_b)$  from the previous section, a natural choice of estimator for  $s_j$  is  $\bar{X}_{bj}/\hat{f}(d_b)$ , for each  $b$ . Under the null hypothesis, for any  $b$ , the expected value of this is approximately  $s_j$ . However, if bin  $b$  contains many true interactions then this expectation no longer holds.

To avoid the influence of true interactions and gain a robust estimator of the  $s_j$ , we take the median as follows:

$$\hat{s}_j = \text{median}_b \frac{\bar{X}_{bj}}{\hat{f}(d_b)}$$

Note that this procedure also follows the DESeq model – specifically, the library size estimation procedure.

## Estimating $s_i$

Our  $s_i$  estimation procedure is broadly similar to the  $s_j$  estimation procedure, but differs in one important aspect.

We have that

$$\mathbb{E}(X_{ij}) = s_j s_i f(d_{ij})$$

Consider a specific other end,  $i$ . An obvious estimator for  $s_i$  is the “normalised” count,  $Y_{ij} = \frac{X_{ij}}{\hat{s}_j \hat{f}(d_{ij})}$ . We could then take the median across  $js$  (i.e. across baits). However, this strategy fails, because we only get information about  $s_i$  for a small number of nearby baits. (Most baits have  $Y_{ij} = 0$  and are therefore not very informative.) Indeed, if some of these baits  $j$  significantly interact with other end  $i$ , then  $s_i$  is greatly overestimated.

To address this, we “pool” other ends together according to how “noisy” they are. However, rather than assuming that noise specifically regresses against some arbitrary choice of explanatory variables, we take a data-driven approach where we postulate that the “noisiness” of each other end is reflected in the number of trans read pairs it is involved in, most of which are noise. Thus, we assign each other end  $i$  to a group  $g(i)$ , according to how many “trans” non-zero counts it has, and whether or not it is also a bait fragment:

$$g(i) = g \Leftrightarrow \begin{cases} \left( \sum_{j:(i,j) \text{ trans}} I(X_{ij} > 0) \right) \in R_g \\ I(i \text{ is a bait}) = b_g \end{cases} \quad (2)$$

where the groups  $R_g$  are defined by the `cut2()` function in the CRAN package *Hmisc* – we ask for 1000 other ends per group, 100 other ends per bait-to-bait group.

It is assumed that all other ends in group  $g$  have approximately the same  $s_i$ . Thus,  $s_i = s_{g(i)}$ .

For each group, we calculate a per-distance bin estimate of  $s_g$ :

$$\bar{Y}_{gb} = \frac{\sum_{i:g(i)=g} \sum_{j;d_{ij} \in B_d} Y_{ij}}{\sum_{i:g(i)=g} \sum_{j;d_{ij} \in B_d} 1}$$

As before, we take the median across these bins:

$$\hat{s}_g = \text{median}_b(\bar{Y}_{gb})$$

## Estimating $r$

We now calculate the dispersion,  $\hat{r}$ . This is simple to obtain from standard NB regression techniques – we find the  $r$  that maximises the likelihood of the regression model:

$$X_{ij} \sim NB(\mu_{ij}, r)$$

Since some of the pairs  $(i, j)$  are true interactions, there is slightly more variance across  $X_{ij}$  than there would be under the null, therefore we expect  $\hat{r}$  to be a slight underestimate of  $r$ . However, the number of interactions is very small compared to the number of pairs (we typically call around 1 – 2% of pairs with  $d_{ij} < 1.5\text{mb}$ ), and thus this effect should be negligible. In any event, underestimation of  $r$  cannot introduce any false positives.

## 5 Estimation – Technical noise

Technical noise is assumed distance-invariant:

$$T_{ij} \sim \text{Pois}(\lambda_{ij})$$

The parameters  $\lambda_{ij}$  are estimated using trans pairs, since there is no contribution from Brownian noise, and thus  $X_{ij} \approx T_{ij}$ .

Because we have little information on  $\lambda_{ij}$ , we again pool fragments together, using a similar rationale to the  $s_i$  estimation procedure. Other ends

get classes  $g(i)$  as before (Equation 2). However, this time, each bait  $j$  also gets a class  $h(j)$  – as in Equation 2, the class is based on the number of other ends the bait interacts with in trans.

We assume that  $\lambda_{ij}$  depends only on the classes of  $i$  and  $j$  –

$$\lambda_{ij} = \Lambda(g(i), h(j))$$

To estimate  $\Lambda$ , we obtain all trans pairs with the appropriate class membership:

$$Q_{g,h} = \{(i, j); (i, j) \text{ trans}, g(i) = g, h(j) = h\}$$

and calculate

$$\Lambda(g, h) = \frac{\sum_{(i,j) \in Q_{g,h}} X_{ij}}{|Q_{g,h}|}$$

## 6 Calculating $p$ -values

Putting the previous sections together we have that, under the null,  $X_{ij}$  has Delaporte distribution:

$$X_{ij} \sim NB(\mu_{ij}, r) + Pois(\lambda_{ij})$$

We perform a simple one-sided location test:

$$H_0 : \mathbb{E}(X_{ij}) = \mu_{ij} + \lambda_{ij}$$

$$H_1 : \mathbb{E}(X_{ij}) > \mu_{ij} + \lambda_{ij}$$

This test is performed using the *Delaporte* package, obtaining  $p$ -values  $p_{ij} = p(X_{ij} \geq x_{ij})$ .<sup>†</sup> CHiCAGO reports these  $p$ -values on the natural logarithmic scale.

---

<sup>†</sup>In some rare situations,  $x_{ij}$  was too large compared to  $E(X_{ij})$ , and we encountered underflow issues. Here, we approximated  $X_{ij}$  by a Negative Binomial distribution, using the Method of Moments. In other words, we assume that, under the null,

$$X_{ij} \approx X'_{ij} \sim NB(\eta_{ij}, \rho_{ij})$$

where  $\eta_{ij}$  and  $\rho_{ij}$  are found by equating the mean and variance of  $\mathbb{E}(X)$  and  $\mathbb{E}(X')$ .

## 7 Working with multiple replicates

We now consider the situation where multiple biological replicates are analysed simultaneously. The replicates are indexed by  $k = 1, \dots, K$ . Thus,  $X_{ijk}$  is the count for other end  $i$ , bait  $j$ , replicate  $k$ .

We obtain sample-specific scaling factors  $s_k$ , in a manner akin to that of DESeq (ANDERS and HUBER, 2010) by looking at regions proximal to baits. The procedure is:

- Take a window around each bait (by default, 1.5mb in either direction)
- Count number of reads, divide by number of other ends present, to get  $M_{jk}$ .
- Take geometric mean across samples.  $G_j = geo_k(M_{jk})$
- $s_k = median_j(M_{jk}/G_j)$

A summarised count is calculated as a weighted average of the individual samples' counts:

$$X_{ij} = round \left( \frac{\sum_k s_k X_{ijk}}{\sum_k s_k} \right)$$

This has parallels to pooling biological replicates (as has been common in ChIP-seq data, for example), but using a more appropriate estimator of library size than simply taking the total number of reads.

This summarised count is taken forward in the analysis.

We can also derive normalised counts  $\tilde{X}_{ijk} = \frac{1}{s_k} X_{ijk}$ , which can be useful for visualisation purposes.

## 8 Multiple testing and $p$ -value weighting

We expect far more interactions to occur at short ranges than at long ranges. However, suppose that we call interactions by applying a threshold directly to  $p$ -values. Of the hypotheses we test, a large majority are long-range interactions. Thus, with more opportunities to return a  $p$ -value below the threshold by chance, our output is dominated by erroneous long-range calls. Another way to look at this is that, when ordering  $p$ -values, there are sufficiently

many long-range interactions with lower  $p$ -values than true short-range interactions that we cannot call the short-range interactions without accepting the long-range false positives as well.

Standard multiple testing procedures fail to address this problem. For example, the Bonferroni and Benjamini-Hochberg methods both choose a stringent  $p$ -value threshold – as described above, this may discard the long-range false positives, but we also lose many short-range true positives.

A number of relevant approaches are described in GUI *et al.* (2012). For example, SUN *et al.* (2006) use a two-population approach, which we could apply by splitting our hypotheses in two using a distance threshold. However, this method is very sensitive to the choice of distance threshold. Moreover, it also assumes a sudden change of behaviour, which is not biologically plausible as there appears to be a more gradual change in behaviour. Thus, we chose the GENOVESE *et al.* (2006) approach,  $p$ -value weighting, which is a generalized version of SUN *et al.* (2006).

We also considered the use of an empirical Bayes treatment, where a prior probability is used to quantify the two behaviours. However, the Bayesian approach requires explicit assumptions of the read distribution under the alternative hypothesis, over and above requiring a larger mean.  $p$ -value weighting can be viewed as a simplified version of an empirical Bayesian treatment, using a “weight” in place of a prior probability. This method circumvents the need to make an arbitrary choice of the prior distribution of read counts under the alternative hypothesis.

The aim of the  $p$ -value weighting strategy is to “upweight” the significance of proximal pairs and “downweight” distal/trans pairs. Using the notation in GENOVESE *et al.* (2006), we make prior “guesses”  $U_{ij}$ . We allow  $U_{ij}$  to depend on  $d_{ij}$ , assuming that short-range interactions are more likely than long-range interactions, with a smooth transition between the two.

Specifically, we assume a bounded logistic regression model – thus,  $U_{ij}$  is assumed a function of  $d_{ij}$  and the vector of parameters  $\Theta = (\alpha, \beta, \gamma, \delta)$ , as follows:

$$U_{ij} = \eta_{ij}U_{max} + (1 - \eta_{ij})U_{min}$$

where:

$$\eta_{ij} = \text{expit}(\alpha + \beta \log(d_{ij})) = \frac{e^{\alpha + \beta \log(d_{ij})}}{1 + e^{\alpha + \beta \log(d_{ij})}}$$

$$U_{min} = \text{expit}(\gamma)$$

$$U_{max} = \text{expit}(\delta)$$

The above depends on the unknown parameter vector,  $\Theta$ . We estimated a default value for  $\Theta$  by producing a “high-confidence” data set, from 7 macrophage samples. To make the procedure more robust against any large scale interactions, we partitioned the data into 5 subsets, estimate  $\Theta$  separately for each subset, then take the final estimate  $\Theta$  as the component-wise median of the subset  $\Theta$ s. If a user wishes to use CHiCAGO on cells whose interactomes are expected to differ greatly, this analysis can be redone to obtain new  $\Theta$ .

To obtain the weights, we first need to calculate  $\bar{U}$ , the mean value of  $U_{ij}$ :

$$\begin{aligned}\bar{U} &= \frac{1}{m} \sum_i \sum_j U_{ij} \\ &= \bar{\eta} U_{max} + (1 - \bar{\eta}) U_{min}\end{aligned}$$

Hence, we calculate the weights  $W_{ij}$  as follows:

$$\begin{aligned}W_{ij} &= \frac{U_{ij}}{\bar{U}} \\ &= \frac{\eta_{ij} U_{max} + (1 - \eta_{ij}) U_{min}}{\bar{\eta} U_{max} + (1 - \bar{\eta}) U_{min}} \\ &= \frac{\eta_{ij} U_{rel} + (1 - \eta_{ij})}{\bar{\eta} U_{rel} + (1 - \bar{\eta})}\end{aligned}$$

leading to weighted  $P$ -values:

$$Q_{ij} = \frac{P_{ij}}{W_{ij}}$$

CHiCAGO reports these values on the log-scale.

GENOVESE *et al.* (2006) obtain a false discovery rate, by applying the Benjamini-Hochberg procedure to their weighted  $p$ -values. Unfortunately, the requirements required for Benjamini-Hochberg are not satisfied: since our data are discrete, we do not have uniform  $p$ -values under the null hypothesis. Thus, our preferred strategy is to set a threshold on the  $Q_{ij}$  values.

To aid interpretation, we also compute a score based on the  $q$ -value as follows:

$$score_{ij} = \max(0, -\log Q_{ij} - \log W_{max})$$

where  $W_{max}$  is the value that  $W_{ij}$  would take when  $d_{ij} = 0$ .

In other words, the score is non-negative, and a positive score occurs only when the evidence for an interaction exceeds that of a proximal pair with no reads.

For most users' analyses, the score will be the most appropriate quantity to threshold on.

## References

- ANDERS, S., and W. HUBER, 2010 Differential expression analysis for sequence count data. *Genome Biology* **11**: R106.
- GENOVESE, C. R., K. ROEDER, and L. WASSERMAN, 2006 False discovery control with p-value weighting. *Biometrika* **93**: 509–524.
- GUI, J., T. D. TOSTESON, and M. BORSUK, 2012 Weighted multiple testing procedures for genomic studies. *BioData mining* **5**: 4.
- SUN, L., R. V. CRAIU, A. D. PATERSON, and S. B. BULL, 2006 Stratified false discovery control for large-scale hypothesis testing with application to genome-wide association studies. *Genetic Epidemiology* **30**: 519–530.

## A Mean bin value

Claim:

$$\mathbb{E}(\bar{X}_{bj}) = s_j f(d_b) + \mathcal{O}(\log(s_i)^2)$$

Proof:

$$\begin{aligned} \mathbb{E}(\bar{X}_{bj}) &= \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} \mathbb{E}(X_{ij}) \\ &= s_j f(d_b) \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} s_i \end{aligned} \tag{3}$$

From a Taylor expansion:

$$\begin{aligned}
& \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} s_i \\
&= \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} e^{\log(s_i)} \\
&= \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} \{1 + \log(s_i) + \mathcal{O}(\log(s_i)^2)\} \\
&= \frac{1}{n_{bj}} \sum_{i; d_{ij} \in B_b} \{1 + \mathcal{O}(\log(s_i)^2)\} \\
&= 1 + \mathcal{O}(\log(s_i)^2) \tag{4}
\end{aligned}$$

where the penultimate step is an application of Equation (1), and the final step uses the definition of  $n_{bj}$ , namely  $n_{bj} = \sum_{i; d_{ij} \in B_b} 1$ .

Plugging Equation (4) into Equation (3) gives the required result.