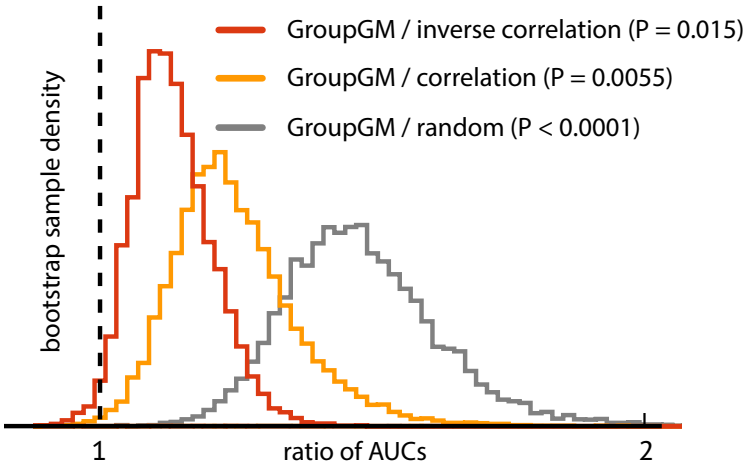
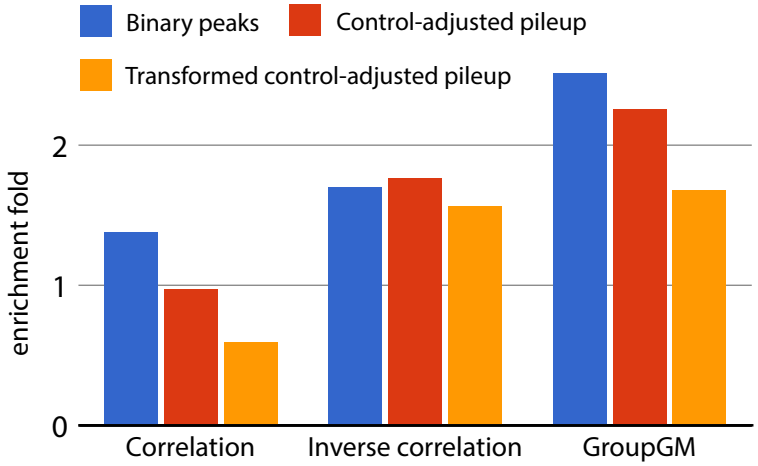
Supplementary information

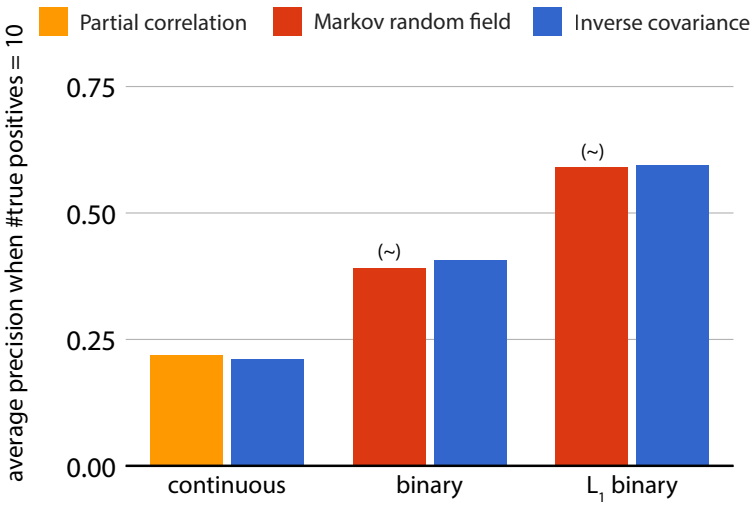
Supplementary figures



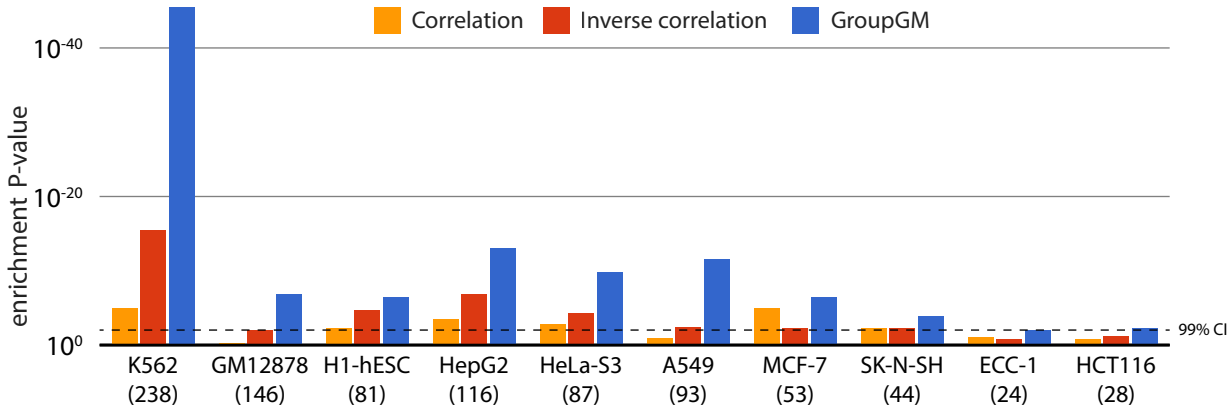
Supplementary Figure 1: Histogram of area under the curve (AUC) ratios comparing enrichment of BioGRID-supported edges in a GroupGM network versus networks created by inverse correlation (red), correlation (yellow), and random edge score assignment (grey). Specifically, we compared the area under enrichment–edge density curves from 10,000 bootstrap samples from chromatin factors, excluding edges between different cell types (Figure 3A). P-values represent the number of bootstrap samples with a ratio of AUC's less than 1.



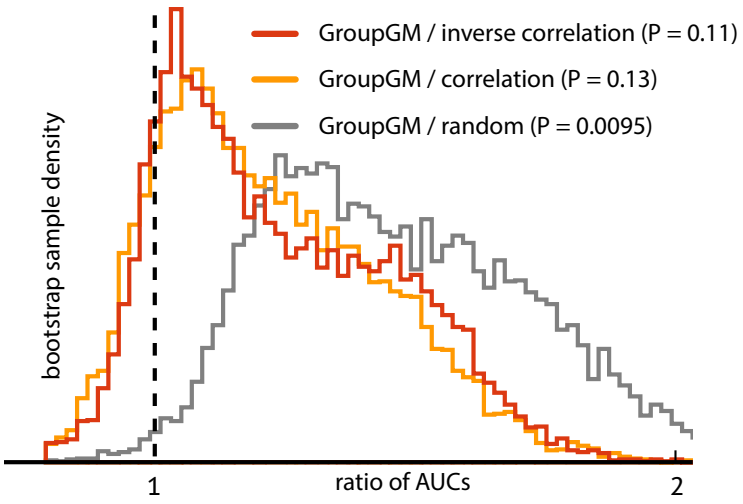
Supplementary Figure 2: Enrichment of BioGRID-supported edges in the K562 cell line of three different modeling approaches using three different pre-processing methods. For binary peaks (blue), we used MACS2 with paired controls and a lenient peak threshold. For control-adjusted pileup (red), we took MACS2 pileup output and adjusted with a paired control. For transformed control-adjusted pileup (yellow), we took the square root of the control-adjusted pileup.



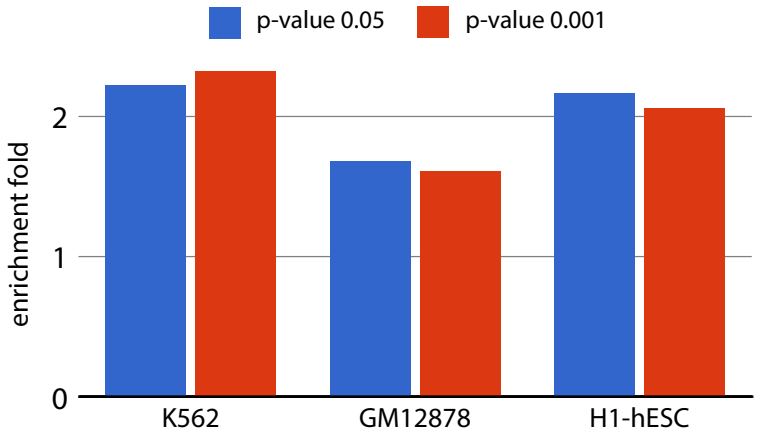
Supplementary Figure 3: Precision when predicting BioGRID interactions using inverse covariance (blue), a binary Markov random field model from [55] (red), and partial correlation (yellow). A tilde (˜) indicates we took Markov random field precision numbers directly from the published precision-recall plot in [55]. To generate inverse covariance and partial correlation results, we started with processed data from [55]. Then, we calculated bootstrap-averaged performance on BioGRID interactions as Zhou et al. did in their article. We compared methods under three different testing regimes. Continuous represents testing on the original control-adjusted, normalized, and binned data. Binary represents testing on binarized data, without regularization. L_1 binary represents testing on binarized data, with L_1 regularization of both models.



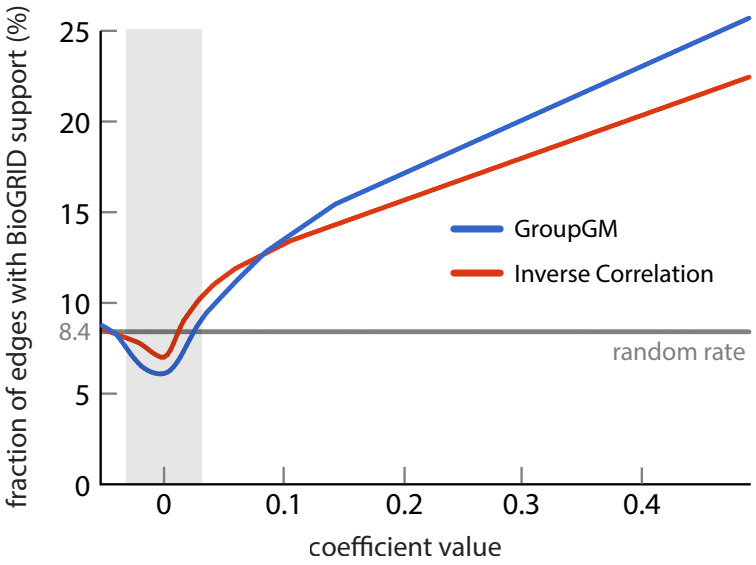
Supplementary Figure 4: One-sided hypergeometric test negative $\log_{10} P$-values for enrichment of BioGRID-supported edges within cell types that have 25 supported edges or more (Figure 3C). The hypergeometric test is less conservative than the bootstrap approach used in Figure 5. Cell types with more datasets will likely have more significant P-values, since they have more edges to compare. Dashed line indicates 99% confidence level ($P = 0.01$). Beneath each cell type name is the number of datasets in that cell type.



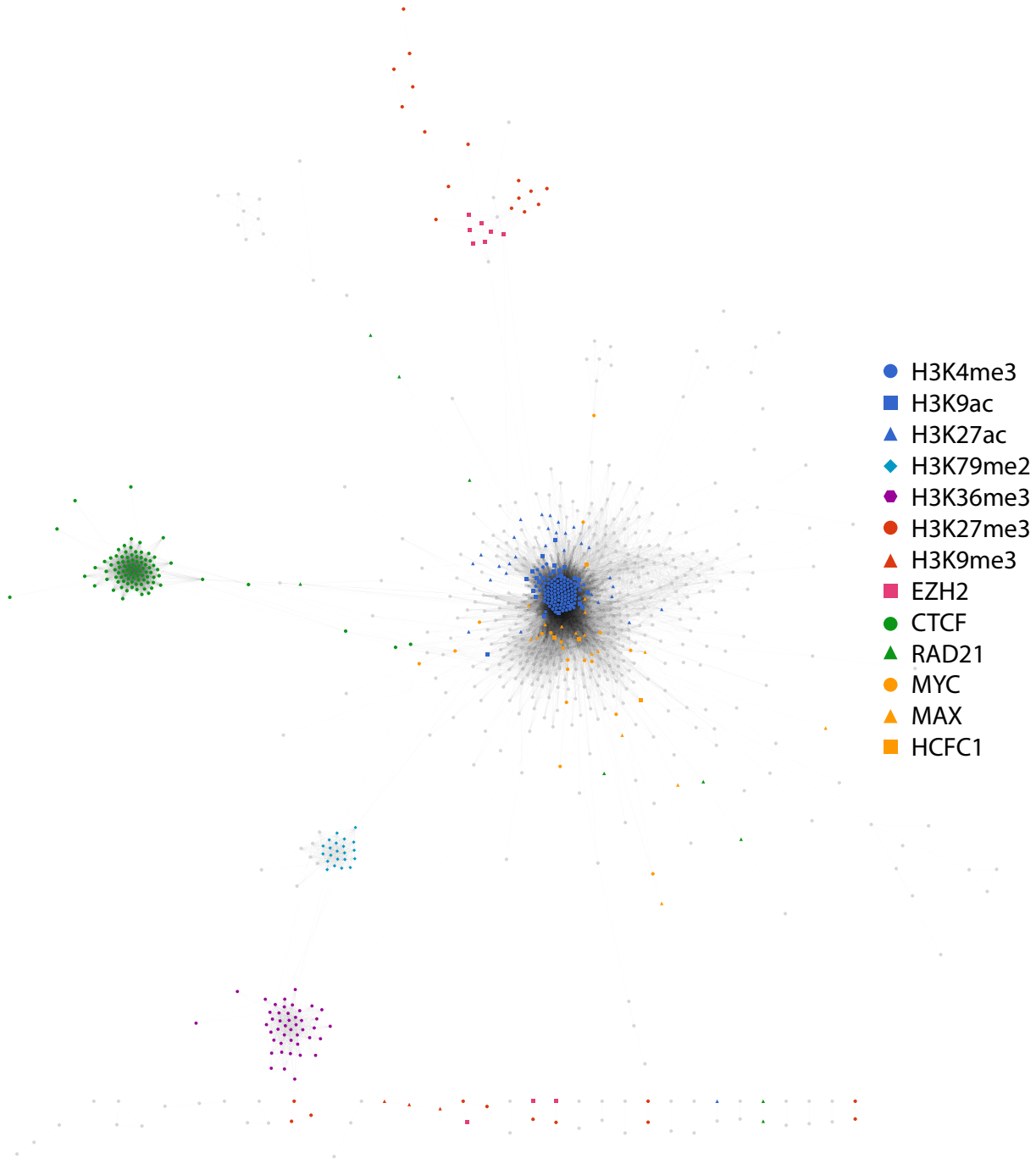
Supplementary Figure 5: Histogram of area under the curve (AUC) ratios comparing enrichment of BioGRID-supported edges in a GroupGM network versus networks created by inverse correlation (red), correlation (yellow), and random assignment (grey). Specifically, we compared the area under enrichment–edge density curves from 10,000 bootstrap samples from chromatin factors, including edges between different cell types (Figure 3B). Variability was much higher than in an examination of edges within cell types (Figure 1). This is because resampling chromatin factors measured in many cell types alters many edges across cell types.



Supplementary Figure 6: Enrichment of BioGRID-supported edges in a GroupGM created from a binary data matrix of MACS peaks called at two different thresholds ($P < 0.05$, blue; $P < 0.001$, red). Within the larger network we examined BioGRID enrichment among datasets from K562 myeloid leukemia cells, GM12878 lymphoblastoid cells, and H1-hESC embryonic stem cells.



Supplementary Figure 7: Enrichment of BioGRID support in edges with a given weight. Negative coefficients indicate negative correlation. Dark grey line indicates the fraction of BioGRID-supported edges in a randomly connected network (8.4%). Light grey shaded area represents those edges with coefficient magnitude less than the 0.03 minimum used in the ChromNet interface.



Supplementary Figure 8: Force-directed 2D embedding embedding of a correlation network of human ENCODE data, estimated by naive correlation. In contrast to the GroupGM network (Figure 6), marginal dependence drives the forces here. Datasets targeting the same chromatin factor are more tightly clustered and spatial relationships between related chromatin factors are much weaker than in Figure 6.

Supplementary tables

Supplementary Table 1: Summary of all ENCODE datasets processed by ChromNet broken down by cell type. This summarizes the full listing of all 1,415 datasets with ENCODE experiment identifiers (Supplementary Data 1). The transcription factor and histone columns represent how many unique transcription factors or histones were measured in that cell type. The treatments column lists the number of additional treatment conditions each cell type was measured under.

Cell type	Datasets	Transcription factors	Histone modifications	Treatments
K562	238	156	12	2
GM12878	146	107	11	1
HepG2	116	82	11	3
A549	93	51	11	2
HeLa-S3	87	64	11	1
H1-hESC	81	60	11	0
MCF-7	53	35	5	1
SK-N-SH	44	27	6	1
endothelial cell of umbilical vein	28	9	12	0
HCT116	28	22	5	0
ECC-1	24	21	0	5
fibroblast of lung	22	2	11	0
keratinocyte	18	2	12	0
mammary epithelial cell	16	2	11	0
SUDHL6	14	2	12	0
Karpas-422	14	2	12	0
CD14-positive monocyte	14	1	11	0
Panc1	13	4	6	0
fibroblast of dermis	13	2	11	0
T-cell acute lymphoblastic leukemia	13	2	11	0
skeletal muscle myoblast	13	2	11	0
astrocyte	13	2	11	0
myotube	13	2	11	0
DOHH2	12	1	11	0
HEK293	12	7	5	0
cardiac mesoderm	12	0	3	0
MCF 10A	12	5	0	2
osteoblast	12	2	10	0
Oci-Ly-1	11	0	11	0
Oci-Ly-3	11	1	10	0
Oci-Ly-7	11	1	10	0
IMR-90	10	10	0	0
NT2/D1	9	3	6	0
GM12891	9	8	0	1
T47D	9	6	0	4
neural cell	8	8	0	0
B cell	8	2	5	0

Cell type	Datasets	Transcription factors	Histone modifications	Treatments
GM12892	7	6	0	1
HL-60	6	5	1	0
PFSK-1	6	5	0	0
NB4	5	4	1	0
U2OS	4	2	2	0
mononuclear cell	4	0	4	0
bronchial epithelial cell	4	1	3	0
foreskin fibroblast	4	1	1	0
kidney epithelial cell	4	1	3	0
GM06990	4	1	3	0
Caco-2	4	1	3	0
BJ	4	1	3	0
LNCaP clone FGC	3	1	1	1
erythroblast	3	2	0	0
WI38	3	1	1	1
cardiac fibroblast	3	1	1	0
H7-hESC	3	0	3	0
H54	2	2	0	0
SH-SY5Y	2	2	0	0
GM08714	2	1	1	0
WERI-Rb-1	2	1	1	0
SK-N-MC	2	1	1	0
epithelial cell of proximal tubule	2	1	1	0
fibroblast of villous mesenchyme	2	1	1	0
retinal pigment epithelial cell	2	1	1	0
fibroblast of pulmonary artery	2	1	1	0
fibroblast of mammary gland	2	1	1	0
HFF-MYC	2	1	1	0
epithelial cell of esophagus	2	1	1	0
choroid plexus epithelial cell	2	1	1	0
cardiac muscle cell	2	1	1	0
brain microvascular endothelial cell	2	1	1	0
astrocyte of the cerebellum	2	1	1	0
astrocyte of the spinal cord	2	1	1	0
GM12875	2	1	1	0
GM12865	2	1	1	0
GM12864	2	1	1	0
BE2C	2	1	1	0
fibroblast of the aortic adventitia	2	1	1	0
fibroblast of skin of abdomen	2	1	1	0
fibroblast of gingiva	2	1	1	0
fibroblast of pedal digit skin	2	1	1	0
fibroblast of upper leg skin	2	1	1	0
GM15510	2	2	0	1
GM19193	2	2	0	1

Cell type	Datasets	Transcription factors	Histone modifications	Treatments
GM18951	2	2	0	1
GM19099	2	2	0	1
GM18505	2	2	0	1
GM18526	2	2	0	1
GM10847	2	2	0	1
Loucy	1	0	1	0
spleen	1	1	0	0
pancreas	1	1	0	0
medulloblastoma	1	1	0	0
lung	1	1	0	0
kidney	1	1	0	0
GM20000	1	1	0	0
GM13977	1	1	0	0
GM13976	1	1	0	0
GM10266	1	1	0	0
GM10248	1	1	0	0
Raji	1	1	0	0
skeletal muscle cell	1	0	1	0
Jurkat	1	0	1	0
GM12874	1	1	0	0
GM12873	1	1	0	0
GM12872	1	1	0	0
GM12801	1	1	0	0
Total	**1,415**	**803**	**353**	**33**

Chromatin factor	Max edge weight	Known in BioGRID
MAX	1.403	+
POLR2A	0.685	+
CTCF	0.358	−
FOS	0.345	−
MXI1	0.322	−
JUND	0.306	−
TBP	0.247	+
MAZ	0.247	+
HCFC1	0.233	−
GTF2F1	0.223	+
EP300	0.187	+
STAT3	0.185	−
E2F6	0.172	−
PHF8	0.172	−
RCOR1	0.167	−
BHLEH40	0.166	−

Supplementary Table 2: Top 16 chromatin factors with a strong connection to MYC in ChromNet. Scores are strongest group edge connecting MYC to the listed factor in any cell type.

Supplementary Note 1: Benefits of binary data

ChIP-seq datasets comprise many sequence reads, and these reads match true chromatin factor locations to varying degrees of quality. Processing of these reads influences the quality of the learned chromatin factor interactions and the computational resources required to rebuild the network with new user-provided data. Binary values representing presence or absence of a chromatin factor at a specific location provided the most effective representation of a ChIP-seq dataset (Supplementary Figure 2).

We compared three different signal representations across three different estimation methods on all K562 datasets. For all methods, we binned the resulting signal into 1,000 bp regions. For the *binary peaks* method, we called peaks from MACS2 with a lenient $P = 0.05$ cutoff. For the *control-adjusted pileup* method, we took quality filtered ($\geq$ level 13) non-multimapping reads and calculated the depth of read pileup in each bin by averaging the depth of the pileup track computed by MACS2 during peak calling. For the *transformed control-adjusted pileup* method, we took the square root of the control-adjusted pileup to make the marginal densities better fit a normal distribution.

The binary peaks method showed the best overall enrichment of BioGRID-supported edges, although the inference method affected performance more than the pre-processing method (Supplementary Figure 2. The binary peaks method likely showed the best performance because it damped noise in large regions without any chromatin factors present. Binary data also vastly reduced data matrix file size. This allows users to download the entire data matrix and add their own datasets.

Supplementary Note 2: Simulation study of estimating a binary Markov random field using the inverse covariance matrix

ChromNet uses an efficient matrix inverse in place of a computationally intensive Markov random field model. An inverse covariance (or correlation) matrix and a Markov random field edge matrix have equivalent sparsity structures if the graph of overlaps between the maximal cliques in the Markov random field graph forms a tree [30]. Trees are a well-known subclass of this set of graphs. However, many networks fall outside this class, specifically those with chordless cycles of four or more nodes. For these graphs the inverse covariance matrix and a Markov random field do not have equivalent sparsity structures [30].

We compared the sparsity structure of a general Markov random field estimated using the inverse covariance matrix against the structure of a Markov random field using node-wise logistic regression. We used node-wise logistic regression because it provides a consistent estimator for Markov random field structure. For comparison with [30], we focus on the inverse covariance matrix. The same observations hold for the inverse correlation matrix used in the main paper, which is just the covariance matrix of normalized data.

A chordless loop of four variables provides the simplest network where the sparsity structures of an inverse covariance matrix and a Markov random field model are not equivalent for binary data. We created such a four-variable Markov random field model, with the same parameters Φ used in [30]:

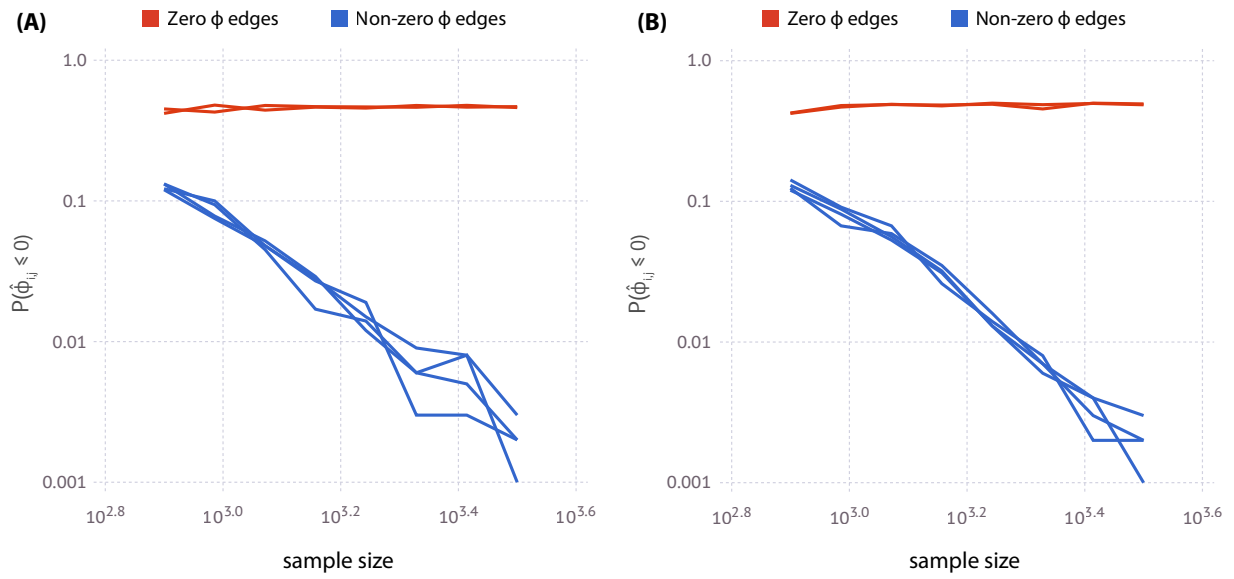
$$\Phi = \begin{pmatrix} 0.1 & 2 & 0 & 2 \\ 2 & 0.1 & 2 & 0 \\ 0 & 2 & 0.1 & 2 \\ 2 & 0 & 2 & 0.1 \end{pmatrix} \tag{3}$$

The entries of this matrix are the parameters to Equation 1 (Methods). Estimating an inverse covariance matrix Ω from an infinite number of samples from the Φ model results in [30]:

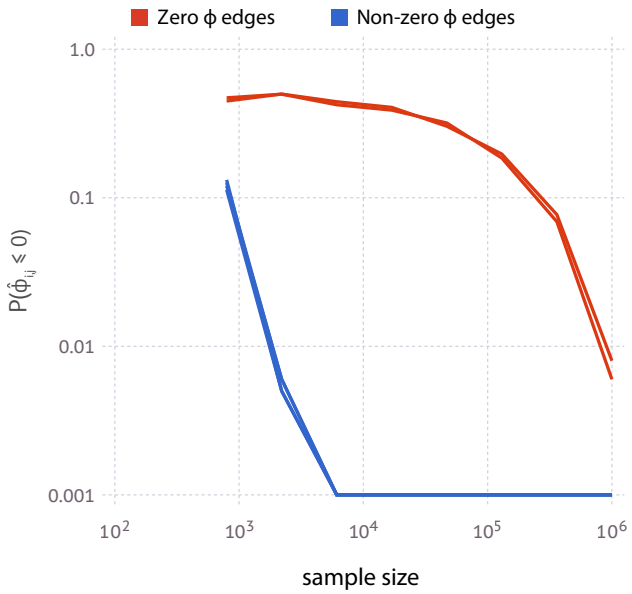
$$\Omega = \begin{pmatrix} 51.37 & -5.37 & -0.17 & -5.37 \\ -5.37 & 51.37 & -5.37 & -0.17 \\ -0.17 & -5.37 & 51.37 & -5.37 \\ -5.37 & -0.17 & -5.37 & 51.37 \end{pmatrix} \tag{4}$$

While Ω and Φ do not have equivalent sparsity structures, the off-diagonal values in Ω that match zeros in Φ are small in magnitude. Only the off-diagonal values matter since we are not including self-self edges in our network. In this case, while asymptotically the sparsity structures do not match, the relative ordering of off-diagonal coefficient magnitudes is fairly consistent. Without regularization, inverse covariance matrix entries will never be exactly zero, so relative magnitude of an entry matters most.

To examine how well the true edges separate from the "false" edges when modeling Φ with an inverse covariance matrix, we calculated $P(\hat{\Phi}_{i,j} \leq 0)$ for each entry in the estimated matrix across a range of sample sizes. We calculated this empirically from the underlying model using 1,000 replicates. This gave an empirical estimate of $P(\hat{\Phi}_{i,j} \leq 0)$ at each sample size. A low value for $P(\hat{\Phi}_{i,j} \leq 0)$ represents a confident detection of a positive edge in the Markov random field. Then, we compared the $P(\hat{\Phi}_{i,j} \leq 0)$ computed using Ω to an equivalent value computed using logistic regression run on each node. That logistic regression is asymptotically consistent with the underlying Markov random field [30]. We seek to use the inverse covariance matrix instead of node-wise logistic regression because GroupGM relies on inverse covariance matrix properties not found in a matrix estimated by node-wise regression.



Supplementary Figure 9: Power to detect positive entries in Φ estimated using **(A)** the inverse covariance matrix or **(B)** node-wise logistic regression. $P(\hat{\Phi}_{i,j} \leq 0)$ represents the probability that an edge is estimated as negative or zero under each method. A small value represents a confident detection of a positive edge. We computed $P(\hat{\Phi}_{i,j} \leq 0)$ empirically by re-running the estimation procedure 1,000 times, while varying the number of samples used to learn the model. Each sample size represents the number of times a sample was drawn from the true network. More samples provides more power to detect positive edges. We plot the estimated $P(\hat{\Phi}_{i,j} \leq 0)$ for the two true zero edges (red) and four true non-zero edges (blue) in Φ.



Supplementary Figure 10: The power the inverse covariance matrix to detect true edges is similar to logistic regression, but unlike logistic regression it eventually identifies false edges as well. The key point to note is the 100-fold separation in power between the true and false edges.

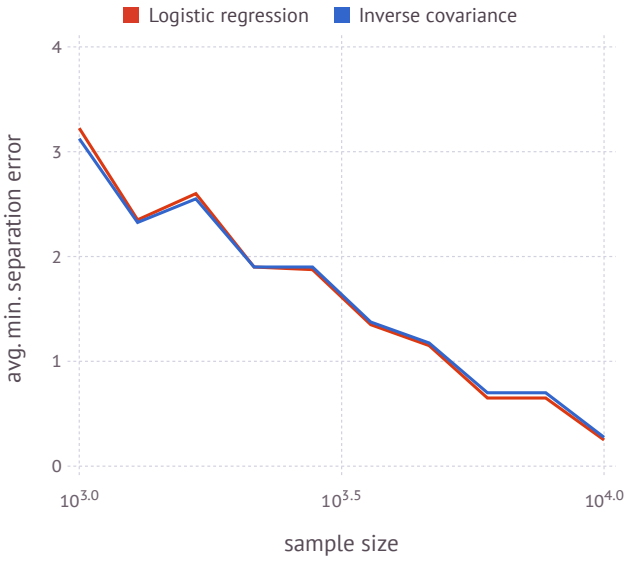
Logistic regression and inverse covariance have very similar power to detect true edges in Φ, although only logistic regression is asymptotically consistent in the considered scenario (Supplementary Figure 9). Supplementary Figure 10 extends the range of sample sizes considered for the inverse covariance matrix estimation. After a 100-fold increase in sample size, false edges not in Φ also begin to be detected, confirming the theoretical inconsistency of the inverse covariance matrix on graphs with chordless cycles. However, the power separation between the true and false edges is very strong. This suggests that, in practice, a proper threshold may be able to separate true from false edges.

When analysing ChIP-seq data in ChromNet we do not test against the null hypothesis of zero edge weight, we instead use a variable threshold controlled by the user, constrained to capture edges enriched for prior interactions (Supplementary Figure 7). We compared the ability of logistic regression and the inverse covariance to separate the true edges from the false edges using a magnitude threshold. This comparison demonstrated nearly equal power between the two methods (Supplementary Figure 11).

While for simulation we compare against logistic regression, we also observe similar performance between a full Markov random field model and inverse covariance in modENCODE data (Supplementary Figure 3, Methods).

Supplementary Note 3: Proof that the group graphical model preserves edge magnitudes in the presence of arbitrary collinearity

The inverse covariance matrix (a symmetric matrix) can be interpreted in terms of multiple regression [22], where for simplicity of notation we assume infinite data samples so $\hat{\Sigma} = \Sigma$:



Supplementary Figure 11: Random 8-node Markov random field models were generated with 40% of the pairwise weight parameters set drawn from $\mathcal{N}(-0.5, 1)$, 1% were set to a large value of 3 to create outliers, and the remaining entries were set to zero. 1,000 such models were drawn and then sampled from. The minimum separation error of true edge from false edges by magnitude was computed for both logistic regression and the inverse covariance matrix across a range of sample sizes. In practive both methods performed equally, even though the inverse covariance matrix is often not asymptotically consistent.

$$\Sigma^{-1} = \Omega = \begin{bmatrix} 1/[\Sigma_{11}(1-R_1^2)] & -\beta_{12}/[\Sigma_{11}(1-R_1^2)] & \cdots & -\beta_{1n}/[\Sigma_{11}(1-R_1^2)] \\ -\beta_{21}/[\Sigma_{22}(1-R_2^2)] & 1/[\Sigma_{22}(1-R_2^2)] & \cdots & -\beta_{2n}/[\Sigma_{22}(1-R_2^2)] \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{n1}/[\Sigma_{nn}(1-R_n^2)] & -\beta_{n2}/[\Sigma_{nn}(1-R_n^2)] & \cdots & 1/[\Sigma_{nn}(1-R_n^2)] \end{bmatrix}$$

where β_{ij} is a parameter of the ith regression that predicts the ith variable from all the others, and R_i^2 is the proportion of the variance in variable i explained by the ith regression. For correlation matrices then on-diagonal Σ_{ii} entries will be one:

$$\Omega = \begin{bmatrix} 1/(1-R_1^2) & -\beta_{12}/(1-R_1^2) & \cdots & -\beta_{1n}/(1-R_1^2) \\ -\beta_{21}/(1-R_2^2) & 1/(1-R_2^2) & \cdots & -\beta_{2n}/(1-R_2^2) \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{n1}/(1-R_n^2) & -\beta_{n2}/(1-R_n^2) & \cdots & 1/(1-R_n^2) \end{bmatrix}$$

To further simplify, we can define $S_i = \frac{-1}{1-R_i^2}$:

$$\Omega = \begin{bmatrix} -S_1 & S_1\beta_{12} & \cdots & S_1\beta_{1n} \\ S_2\beta_{21} & -S_2 & \cdots & S_2\beta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_n\beta_{n1} & S_n\beta_{n2} & \cdots & -S_n \end{bmatrix}$$

Consider an arbitrary edge between two nodes A and B with that correspond to rows A_1 and B_1 in Ω. The strength of the connection in the symmetric matrix Ω is $S_{A_1}\beta_{A_1B_1} = S_{B_1}\beta_{B_1A_1}$.

Now consider a new data set with a superset of the variables in the original network represented by Ω. This new dataset, represented by $\Omega^{(2)}$, has a second B variable with index B_2. These two B variables (B_1 and B_2) are arbitrarily similar to one another but not identical, and the second variable bears no relationship to other variables in the network beyond what it gains by being similar to B_1. The regression problem for A_1 would be unstable, because B_1 and B_2 are highly correlated to each other, which makes it unclear how the weights should be distributed to these two predictor variables. However, the sum of the coefficients for the B group remains the same:

$$\beta^{(2)}_{A_1B_1} + \beta^{(2)}_{A_1B_2} = \beta_{A_1B_1},$$

In addition, no new information has been provided about A, so S_A remains unchanged (because the amount of variance explained remains the same):

$$S^{(2)}_A = S_A,$$

which means the following:

$$S^{(2)}_A \beta^{(2)}_{A_1B_1} + S^{(2)}_A \beta^{(2)}_{A_1B_2} = S_A \beta_{A_1B_1},$$

which is equivalent to:

$$\Omega^{(2)}_{A_1B_1} + \Omega^{(2)}_{A_1B_2} = \Omega_{A_1B_1}.$$

This means that the connection strength that was present in between A and B in Ω is now preserved as a sum of two entries in $\Omega^{(2)}$. This argument generalizes to any number of variables in the B group.

Now after adding a redundant B variable consider adding a redundant A variable to create a new data set $\Omega^{(3)}$. Since the B variables cannot choose between A_1 and A_2 their coefficients are unstable but still sum to their previous value:

$$\beta^{(3)}_{B_1A_1} + \beta^{(3)}_{B_1A_2} = \beta^{(2)}_{B_1A_1} \tag{5}$$
$$\beta^{(3)}_{B_2A_1} + \beta^{(3)}_{B_2A_2} = \beta^{(2)}_{B_2A_1} \tag{6}$$

adding A_2 provided no new explanatory power for the B variables so

$$S^{(3)}_{B_1} = S^{(2)}_{B_1} \tag{7}$$
$$S^{(3)}_{B_2} = S^{(2)}_{B_2}, \tag{8}$$

which means

$$S^{(3)}_{B_1} \beta^{(3)}_{B_1A_1} + S^{(3)}_{B_1} \beta^{(3)}_{B_1A_2} = S^{(2)}_{B_1} \beta^{(2)}_{B_1A_1} \tag{9}$$
$$S^{(3)}_{B_2} \beta^{(3)}_{B_2A_1} + S^{(3)}_{B_2} \beta^{(3)}_{B_2A_2} = S^{(2)}_{B_2} \beta^{(2)}_{B_2A_1}, \tag{10}$$

and

$$\Omega^{(3)}_{B_1A_1} + \Omega^{(3)}_{B_1A_2} = \Omega^{(2)}_{B_1A_1} \tag{11}$$
$$\Omega^{(3)}_{B_2A_1} + \Omega^{(3)}_{B_2A_2} = \Omega^{(2)}_{B_2A_1}. \tag{12}$$

Because Ω is symmetric we know that

$$\Omega^{(2)}_{A_1B_1} + \Omega^{(2)}_{A_1B_2} = \Omega^{(2)}_{B_1A_1} + \Omega^{(2)}_{B_2A_1}.$$

Using this we can now calculate the original connection strength $\Omega_{A_1B_1}$ as a sum of entries in $\Omega^{(3)}$. This can be directly generalized to any number of variables in each group, which means that the connection strength of an edge between two variables in a non-redundant data set can be recovered by summing edges in a data set where both variables are in groups of redundant variables.

$$\Omega_{A_1B_1} = \Omega^{(2)}_{A_1B_1} + \Omega^{(2)}_{A_1B_2} \tag{13}$$

$$\Omega_{A_1B_1} = \Omega^{(2)}_{B_1A_1} + \Omega^{(2)}_{B_2A_1} \tag{14}$$

$$\Omega_{A_1B_1} = \Omega^{(3)}_{B_1A_1} + \Omega^{(3)}_{B_1A_2} + \Omega^{(3)}_{B_2A_1} + \Omega^{(3)}_{B_2A_2} \tag{15}$$

$$\tag{16}$$

Supplementary Note 4: Comparison of Markov random field and inverse covariance for network estimation from binary data

Motivated by the computational advantages of the inverse covariance matrix we compared the performance of both methods applied to binary data from 73 modENCODE ChIP-chip datasets on *Drosophila melanogaster* embryonic S2-DRSC cells from Zhou et al. [55] (Supplementary Figure 3). The authors reported 10 known positives in the top 15 predicted interactions when using an L_1-penalized Markov random field (max entropy) model. We obtained the same performance using L_1-penalized inverse covariance methods (graphical lasso [12]) when choosing a regularization parameter that maximized the precision. Similarly the performance of unregularized estimation was also equivalent between the two models. Partial correlation is a rescaled version of the inverse covariance matrix used by the authors on real valued data. We found it performed similarly to the inverse covariance matrix. For the tested ChIP-chip datasets, using binary data and L_1 regularization shows a clear advantage (Supplementary Figure 3). For ENCODE ChIP-seq data, however, we found a benefit for binarization, but not L_1 regularization. This may be because the human genome is much longer than the fly genome, and so provides many more positional samples to prevent overfitting.