CHROGPS, A GLOBAL CHROMATIN POSITIONING SYSTEM FOR THE FUNCTIONAL ANALYSIS AND VISUALIZATION OF THE EPIGENOME (SUPPLEMENTARY MATERIAL)

JOAN FONT-BURGADA, OSCAR REINA, DAVID ROSSELL, FERNANDO AZORÍN

1. Data acquisition and formatting

We downloaded predicted binding sites for a number of epigenetic factors in *Drosophila melanogaster* from the modENCODE project at `www.modencode.org` (Celniker et al., 2009). Supplementary Tables 1 and 2 provide a complete list of the considered factors in S2 and BG3 cells, respectively.

Specifically, we downloaded the provided GFF files and selected the genomic regions tagged as significantly enriched. These files indicate the chromosome, start and end (in base pairs) for all predicted binding sites of a given factor in the *Drosophila* genome dm3. Our Bioconductor (Gentleman et al., 2004) package `chroGPS` provides routines `getURL` and `gff2RDList` for downloading modENCODE data and formatting it into adequate R objects. For the S2 factors we also downloaded WIG files from modENCODE, which contain smoothed scores for each probeset in the tiling arrays.

The genomic intervals in the GFF files were the raw data for chroGPS-factors with iOverlap, iTanimoto and chi-square distances (Section 3.1). The smoothed scores were the input both for chroGPS-factors with correlation distance (Section 3.1) and for Principal Component Analysis.

For chroGPS-genes we annotated each gene by checking which factors had a binding site within 1kb up or downstream, using the function `annotatePeakInBatch` of the Bioconductor package ChIPpeakAnno v2.4.0 (Zhu et al., 2010) and the Ensembl reference genome (Release 66, February 2012). The input data for chroGPS-genes was a matrix with genes as rows and factors as columns, where cell (i, j) takes the value 1 if gene i has a nearby predicted binding site for factor j.

2. MDS representation quality measures

Classical Multidimensional Scaling (Torgerson, 1952) seeks to minimize a loss function called *stress*, which is simply the sum of squared

differences

$$\sum_{i>j}(d_{ij} - \tilde{d}_{ij})^2, \tag{1}$$

where d_{ij} is the input dissimilarity between objects i and j and $\tilde{d}_{ij}$ is the corresponding Euclidean distance in the low-dimensional representation. The stress (1) is not a satisfactory measure of goodness-of-fit, as it depends on the scale of the original distances. Instead, Kruskal (1964a,b) proposed the *stress-1* function

$$\frac{\sum_{i>j}(d_{ij} - \tilde{d}_{ij})^2}{\sum_{i<j} d_{ij}^2}, \tag{2}$$

which is normalized by the overall magnitude of the dissimilarities and is invariant to scale transformations of d_{ij}. Because the denominator in (2) depends only on the input distances d_{ij}, minimizing (2) is equivalent to minimizing (1). Sibson (1972) proposed using the R^2 coefficient instead, *i.e.* the squared Pearson correlation between d_{ij} and $\tilde{d}_{ij}$. R^2 is invariant to location and scale transformations and guaranteed to be in $[0, 1]$.

While we report both stress-1 and R^2, we favour the latter as the primary quality assessment metric for the properties discussed above and its familiar interpretation as the percentage of variability in d_{ij} captured by a linear function of $\tilde{d}_{ij}$.

3. chroGPS-factors

3.1. **Dissimilarity metrics.** chroGPS-factors requires a similarity measure between pairs of factors. We implement distances for both when the input data are lists of genomic intervals (*e.g.* predicted binding sites) and when they are continuous measurements (*e.g.* microarray probeset intensities or sequencing read coverage).

We consider three metrics to measure genome-wide overlap of genomic intervals: interval Tanimoto (iTanimoto), average interval overlap (iOverlap) and chi-square. iTanimoto is a novel metric that generalizes the Tanimoto similarity to interval data, and measures the size of the intersection between two sets relative to their union. Let (i, j) be a pair of epigenetic factors and N_i, N_j their respective number of intervals (binding sites). Let $a_k = 1$ indicate that interval $k = 1, \ldots, N_i$ in factor i overlaps with some interval in factor j ($a_k = 0$ otherwise), and similarly let $b_k = 1$ indicate that interval $k = 1, \ldots, N_j$ in j overlaps with some interval in i. We define the iTanimoto similarity between i and j as

$$s_{ij}^T = \frac{\frac{1}{2}\left(\sum_{k=1}^{N_i} a_k + \sum_{k=1}^{N_j} b_k\right)}{N_i + N_j - \frac{1}{2}\left(\sum_{k=1}^{N_i} a_k + \sum_{k=1}^{N_j} b_k\right)}, \tag{3}$$

where the numerator measures the overlap between i and j. Notice that if a_k and b_k were elements from binary vectors defined on a common set of variables, then $a_k = 1$ if and only if $b_k = 1$, and (3) would reduce to the usual Tanimoto similarity. As desired, s^T_{ij} has the property that when all intervals in i overlap with some interval in j ($a_i = 1$) and vice versa ($b_i = 1$), then $s^T_{ij} = 1$. Similarly, $s^T_{ij} = 0$ when no intervals overlap with each other.

We transform similarities into distances $d^T_{ij} = 1 - s^T_{ij}$, as it facilitates comparison with clustering, can be directly used with non-metric or weighted scaling strategies and has been reported to provide better solutions in low dimensions than similarity-bases scaling (Buja et al., 2008). Since $s^T_{ii} = s^T_{jj} = 1$, our conversion is essentially equivalent to the square root of the usual identity $d^2_{ij} = (s_{ii} + s_{jj} - 2s_{ij}) = 2(1 - s_{ij})$ between Euclidean similarities (inner products) and distances. We favored the definition $d^T_{ij} = 1 - s^T_{ij}$ for its clearer interpretation: $d^T_{ij} = 0$ corresponds to perfect overlap, $d^T_{ij} = 0.5$ to 50% overlap and $d^T_{ij} = 1$ to no overlap.

A property of iTanimoto is that it tends to assign more weight to epigenetic factors with a larger number of binding sites. Intuitively, when $N_i >> N_j$ most intervals in i do not overlap an interval in j ($a_k = 0$) and s^T_{ij} in (3) tends to be small. This behavior holds even when all j intervals overlap with i ($b_i = 1$), as then the numerator approaches $0.5N_j << N_i$, giving $s^T_{ij} \approx 0$. To address this issue we defined the average interval overlap (iOverlap) as

$$s^o_{ij} = \frac{1}{2}\left(\frac{\sum_{i=1}^{N_i} a_k}{N_i} + \frac{\sum_{i=1}^{N_j} b_i}{N_j}\right), \tag{4}$$

i.e. the average between the proportion of i intervals included in j intervals and j intervals included in i intervals. The definition guarantees that $s^o_{ij} > 0.5$ whenever all j intervals are included in an i interval (or *vice versa*), even when $N_i >> N_j$.

Finally, we also considered the chi-square distance, as it is a classical metric used for categorical data dimensionality reduction techniques. Here we defined $a_k = 1$ if the k^{th} base in the genome is covered by an i interval ($a_k = 0$ otherwise), and similarly for b_k. As usual, the chi-square distance d^c_{ij} is the chi-square test statistic for testing the statistical association between a_k and b_k. Hence d^c_{ij} is unbounded (unlike d^T_{ij} and d^o_{ij}), which can cause the global shape of the map to be dominated by large distances.

For continuous measurements we considered two approaches. First, we used Principal Component Analysis (PCA). Secondly, we computed the Pearson correlation r_{ij} between genome-wide profiles and define their distance as $d_{ij} = (1 - r_{ij})/2$.

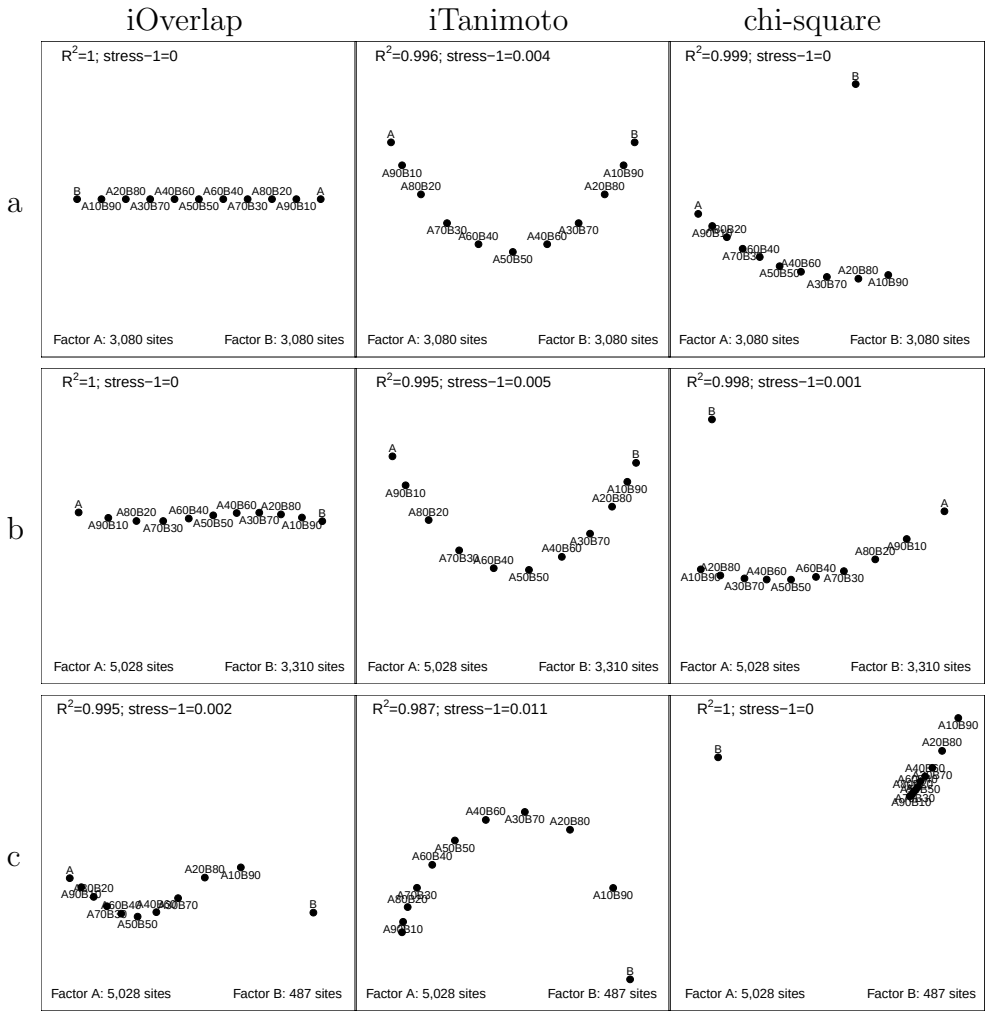
3.2. **Metrics performance assessment.** We studied the performance of the iTanimoto, iOverlap and chi-square metrics (Section 3.1) in synthetic and experimental data. For the simulation study we considered a simplified scenario with a dilution series. We defined two factors with no overlapping binding sites and produced a dilution series by combining a certain proportion of sites from each factor. In order to keep the simulation realistic, we chose the factors from the modEncode S2 data (Supplementary Table 1). The specific steps are indicated in Algorithm 1 below.

Algorithm 1. Dilution series simulated study

(1) Select two factors A and B from the S2 data. Remove all binding sites that overlap between A and B to create pure factors.
(2) Generate a dilution series with 9 artificial factors. First, create a factor by selecting 90% random peaks from A and 10% from B. Then create a factor with 80%/20%, and so on up to 10%/90%.
(3) Compute pairwise distances between all factors.
(4) Produce a 2-dimensional MDS map.

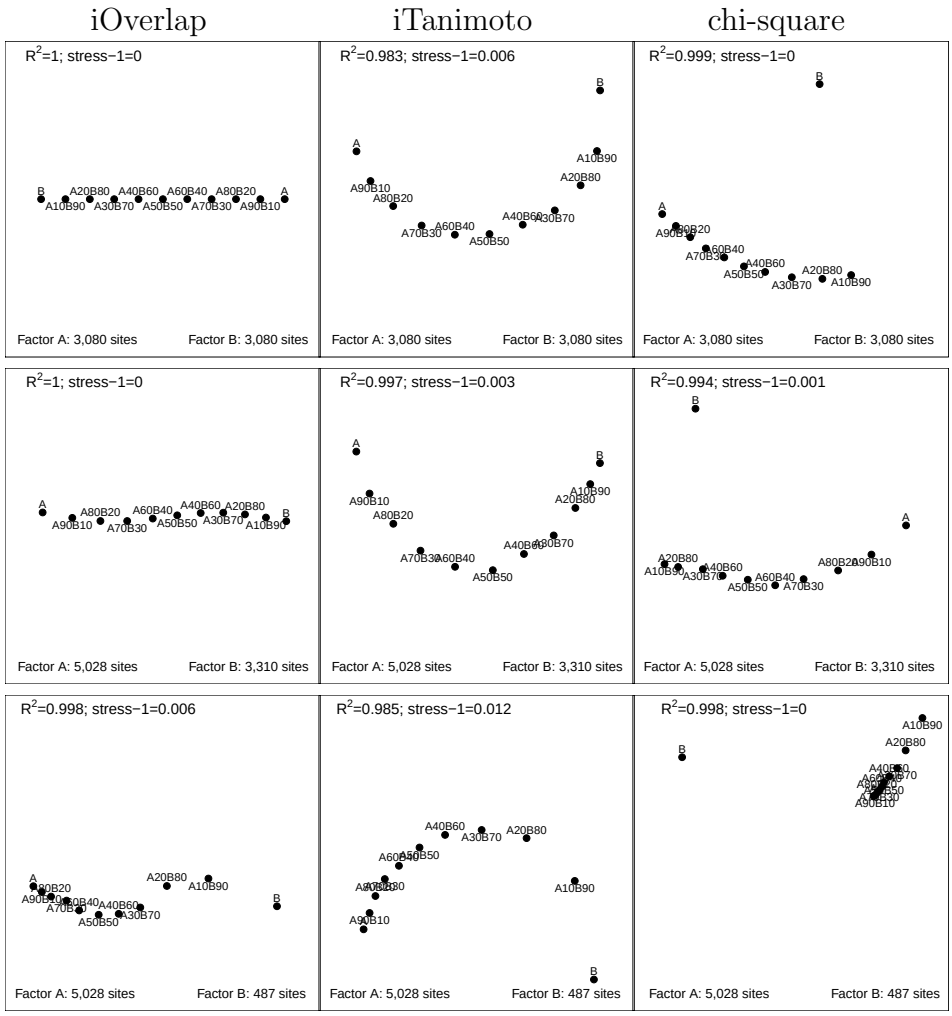
In this simplified scenario, the ideal map would represent the dilution series equidistantly along a straight line connecting the two factors. Supplementary Figures 1 and 2 show the maps generated with classical scaling and isoMDS (Venables and Ripley, 2002), respectively. Panels a show maps for factors A=H3K27Me3 and B=HP1b, which have a similar number of binding sites (3,310 and 3,080, respectively). We equaled the number of sites in each factor by selecting 3,080 random sites from A. The dilution series appears right in its idealized locations for the iOverlap metric. For iTanimoto the series is located in equidistant points along a half circle, while for chi-square factors A and B are treated in a highly asymmetric manner. The latter metric measures distances at the base pair level and is thus sensitive to the binding sites width, which is undesirable in our setup. Panels b in Supplementary Figures 1-2 show factors A=H3K27Me3 and B=H3K36Me3, which differ in the number of binding sites (3,291 and 5,028, respectively). Points show a minor departure from their idealized locations for iOverlap, while iTanimoto and chi-square present similar configurations as before. Panels c show an extreme case where factor A=H3K36Me3 has roughly ten times more binding sites than factor B=EZ (5,007 and 487, respectively). All metrics are affected by this deep imbalance, but again iOverlap results in a representation closest to an equi-distant straight line between A and B. Chi-square induces a highly skewed map where the larger distances dominate the map configuration.

Supplementary Figure 3 shows the chroGPS-factors maps obtained with the three metrics in the modEncode S2 data and isoMDS, as well as with PCA and the Pearson correlation based isoMDS. iOverlap and iTanimoto result in very similar maps, both separating clearly the four



Supplementary Figure 1. Classical MDS maps for simulated dilution experiments. a: H3K27Me3, HP1b; b: H3K27Me3, H3K36Me3; c: H3K36Me3, EZ

functional domains. Matching our simulation study results, chi-square distances tend to give more weight to factors with fewer or narrower binding sites, *e.g.* individual replicates of EZ, PSC2, and SU(VAR)39 with 487, 449 and 367 sites (respectively). The two EZ replicates appear closeby in the iOverlap and iTanimoto maps, but substantially separated in the chi-square map. The asymmetric distribution of points induced by the chi-square metric hampers the interpretation of denser areas of the map. PCA results in a map which explains moderate variability and fails to clearly separate domains, making straightforward interpretation difficult. The Pearson correlation map delivers a similar configuration and R^2 to those of iOverlap and iTanimoto, but functional domains are not as well conserved in some cases. iOverlap and

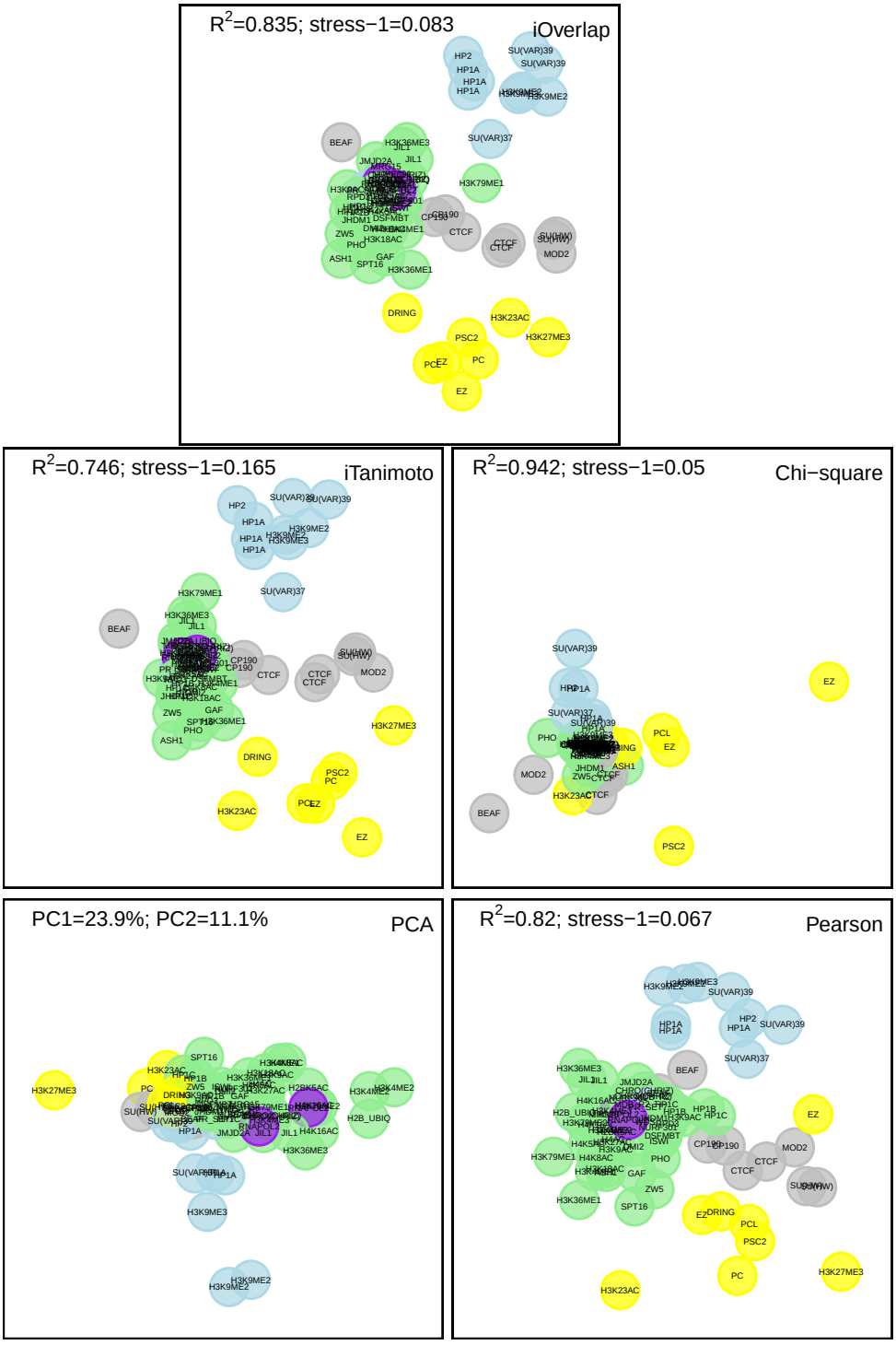


SUPPLEMENTARY FIGURE 2. isoMDS maps for simulated dilution experiments. a: H3K27Me3, HP1b; b: H3K27Me3, H3K36Me3; c: H3K36Me3, EZ

iTanimoto provide essentially the same information, but we decided to use iOverlap for its better simulation results and higher R^2 and stress-1 values.

4. chroGPS-genes

4.1. **Dissimilarity metrics.** The input data for chroGPS-genes is a $n \times p$ binary matrix X with entry $x_{ik} = 1$ if gene $i = 1, \ldots, n$ has a nearby predicted binding site for factor $k = 1, \ldots, p$ (see Section 1 for details). Here measuring distances between genes is equivalent to measuring distances between binary vectors. We considered 4 metrics: chi-square, Tanimoto, Weighted Tanimoto and average overlap. Chi-square distances are the classical choice used by correspondence



SUPPLEMENTARY FIGURE 3. isoMDS of chroGPS-factors maps for S2 genomic binding site information using iOverlap, iTanimoto and Chi-square distances. PCA and Pearson correlation MDS produced using continous measurements for S2 data.

analysis, a popular dimensionality reduction technique for binary matrices (Benzécri, 1973). Because of their being unbounded, the map may be dominated by large distances. The three remaining metrics have the attractive property of being bounded between 0 and 1.

The Tanimoto coefficient measures the proportion of common factors relative to the total number of factors. The usual Tanimoto coefficient gives unduly large weight to factors with several replicates (*e.g.* profiles were obtained using several antibodies or technologies). To address this issue, we extended the Tanimoto coefficient so that each factor receives the same overall weight, and this weight is split evenly across its replicates (*e.g.* for a factor with 2 replicates we assign weight 1/2 to each, with 3 replicates 1/3, etc). More precisely, column k in X is assigned a weight $w_k = 1/r_k$, where r_k is the number of replicates for the corresponding factor. Let $N_{i.} = \sum_{k=1}^{p} w_k x_{ik}$ be the (weighted) number of factors assigned to gene i, $N_{j.}$ be that for gene j and $N_{ij} = \sum_{k=1}^{p} w_k x_{ik} x_{jk}$ the number of common factors. The Tanimoto similarity measure is

$$s_{ij}^{T} = \frac{N_{ij}}{N_{i.} + N_{j.} - N_{ij}}. \tag{5}$$

We define the average overlap metric analogously to (4) as

$$s_{ij}^{o} = \frac{1}{2}\left(\frac{N_{ij}}{N_{i.}} + \frac{N_{ij}}{N_{j.}}\right). \tag{6}$$

That is, s_{ij}^{o} is the average between the proportion of i factors shared by j and j factors shared by i.

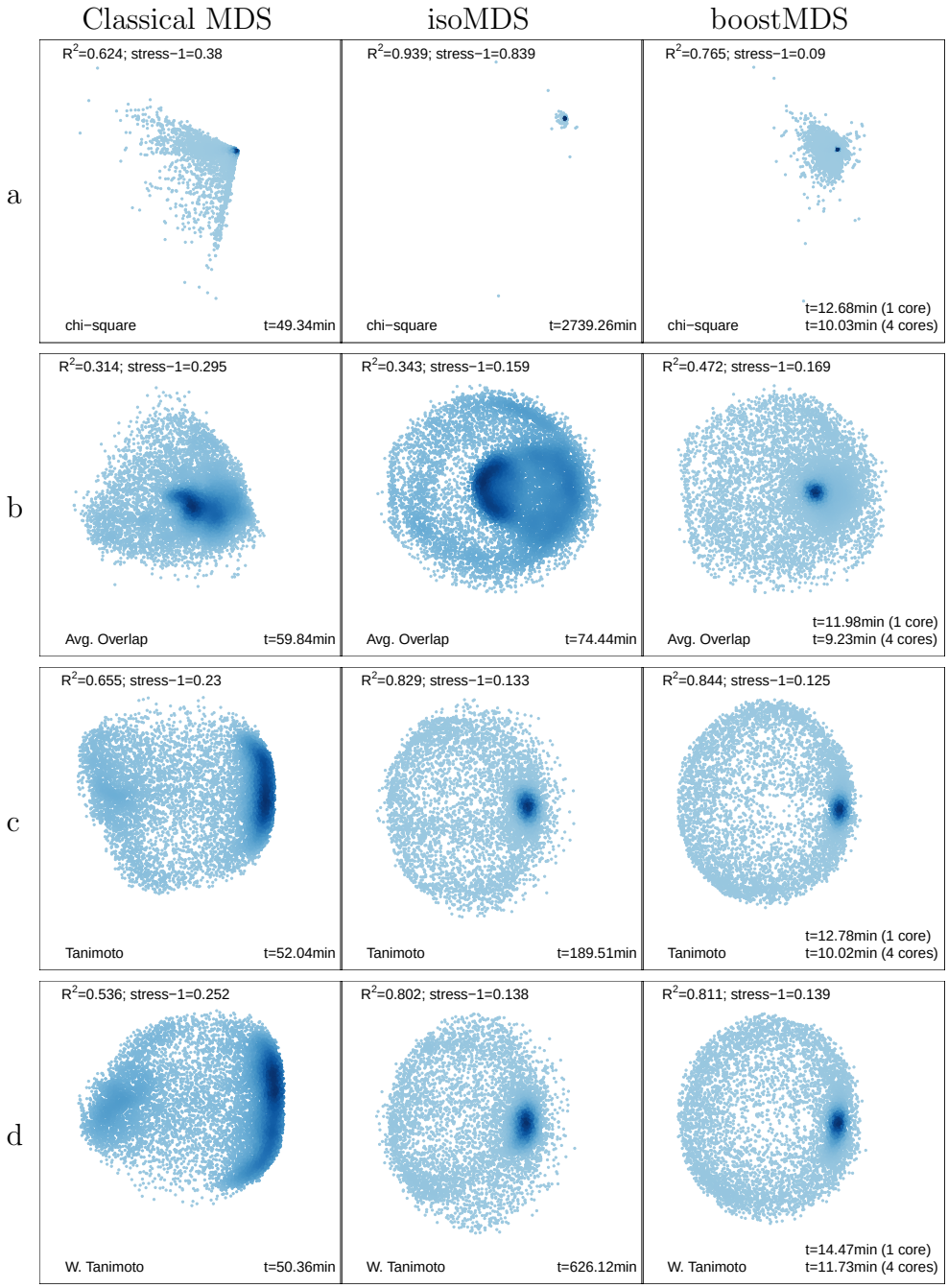
Both Tanimoto and average overlap weight all factors equally. We hypothesized that this property might not always be reasonable from a biological point of view, *e.g.* sharing a scarce factor might be more biologically informative than sharing a frequent one. To account for this possibility, we defined the Weighted Tanimoto similarity by assigning weights inversely proportional to the (square root) number of genes with that factor (akin to the chi-square metric). Let $N_{.k} = \sum_{i=1}^{n} x_{ik}$ be the number of genes with factor k. The Weighted Tanimoto similarity between i and j is defined as

$$s_{ij}^{WT} = \frac{\sum_{k=1}^{p} \frac{1}{\sqrt{N_{.k}}} \mathrm{I}(x_{ik} = 1)\mathrm{I}(x_{jk} = 1)}{\sum_{k=1}^{p} \frac{1}{\sqrt{N_{.k}}} \mathrm{I}(x_{ik} + x_{jk} > 0)}, \tag{7}$$

where I() is the indicator function.

As in chroGPS-factors, we transform similarities into distances by taking $d_{ij}^{T} = 1 - s_{ij}^{T}$, $d_{ij}^{o} = 1 - s_{ij}^{o}$ and $d_{ij}^{WT} = 1 - s_{ij}^{WT}$.

4.2. **Metrics performance assessment.** Similar to what we observed for chroGPS-factors (Section 3.2), the chi-square distance tends to produce chroGPS-genes maps with a highly non-uniform distribution.



SUPPLEMENTARY FIGURE 4. chroGPS-genes maps obtained with classical MDS, isoMDS and boostMDS for chi-square (a), average overlap (b), Tanimoto (c) and Weighted Tanimoto (d).

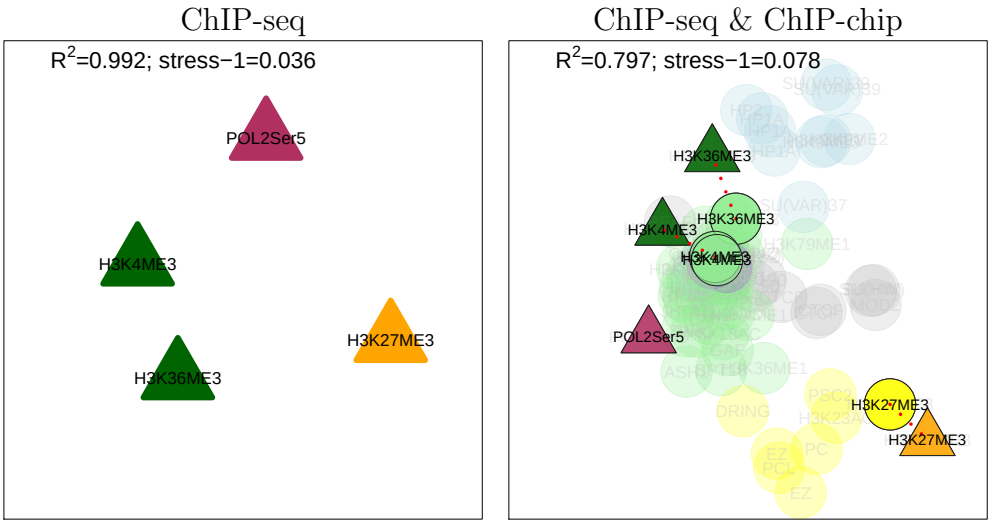
Supplementary Figure 4 shows *Drosophila melanogaster* S2 maps obtained with several MDS algorithms (see Section 6 for details). The chi-square map focused on representing accurately a few outlying genes (*i.e.* genes with large distance to most other genes), concentrating the remaining genes in areas of high density and difficult interpretation. Additionally, the representations achieved modest values for the R^2 and stress-1 functions (Section 2).

Average overlap (Supplementary Figure 4b) produced less skewed maps but achieved relatively poor R^2 and stress-1 values. Tanimoto and weighted Tanimoto produced maps with similar configurations that are amenable to useful biological interpretation (see the manuscript for details), and achieved fairly high R^2 and stress-1. We used Tanimoto distances for our final analysis, given their good performance and familiar definition.

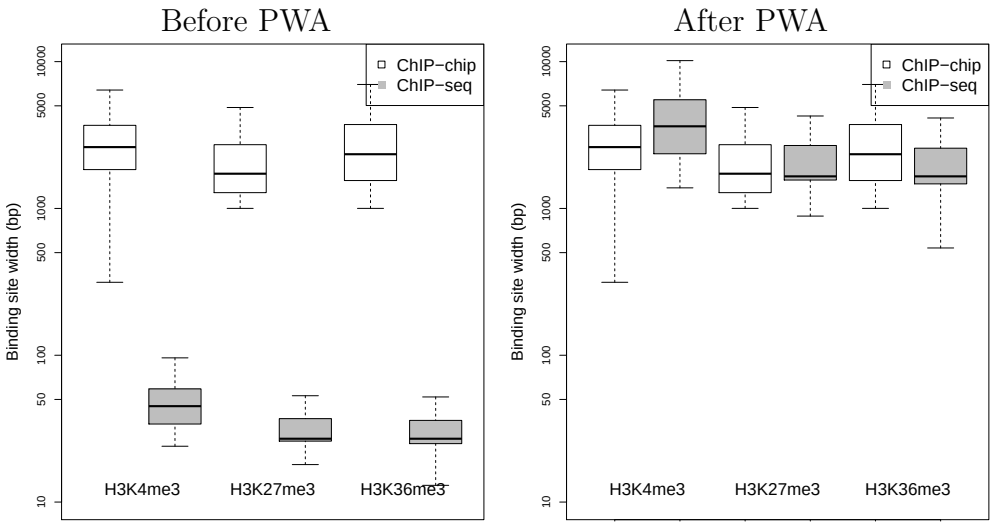
5. Data integration and conservation assessment

We anticipate two main uses for merging multiple data sets into a single map. A first obvious application is integrating results from different technologies or experimental procedures in order to obtain an overall view. Another relevant use is assessing conservation across genetic backgrounds, *e.g.* factor-factor or gene-factor associations. For both uses we adopt a two-step procedure. First, we generate a joint map by computing a distance matrix including points from all data sets and represent it via MDS. Second, we correct for systematic differences between data sources by using an adequate adjustment method. In Section 5.1 we propose several adjustment strategies and discuss their relative merits. In Section 5.2 we discuss the use of combined maps, and the role of adjustment strategies therein, to assess conservation across genetic backgrounds or conditions.

5.1. **Integration into joint maps.** We found that directly generating a joint map was preferable to producing separate maps for each data set and later combining them. As an illustration, we considered integrating S2 ChIP-chip data for 76 factors with ChIP-seq data for 4 factors into a joint chroGPS-factors map. For 3 out of these 4 factors (H3K4me3, H3K27me3, H3K36me3) we also had ChIP-chip data available. Producing a separate map for the ChIP-seq samples resulted in a map with limited contextual information (Supplementary Figure 5). On the other hand, a joint map located each factor relatively close to its ChIP-chip counterpart. More importantly, the relative distances between ChIP-seq samples changed to take into account the more extensive information contained in the ChIP-chip data. While the joint map already provides useful information, ChIP-seq samples form an external layer around the ChIP-chip samples, *i.e.* they exhibit a systematic bias.



SUPPLEMENTARY FIGURE 5. isoMDS S2 chroGPS-factors map. Dots connect ChIP-seq and ChIP-chip samples



SUPPLEMENTARY FIGURE 6. Peak width distribution for S2 ChIP-chip/ChIP-seq replicated factors

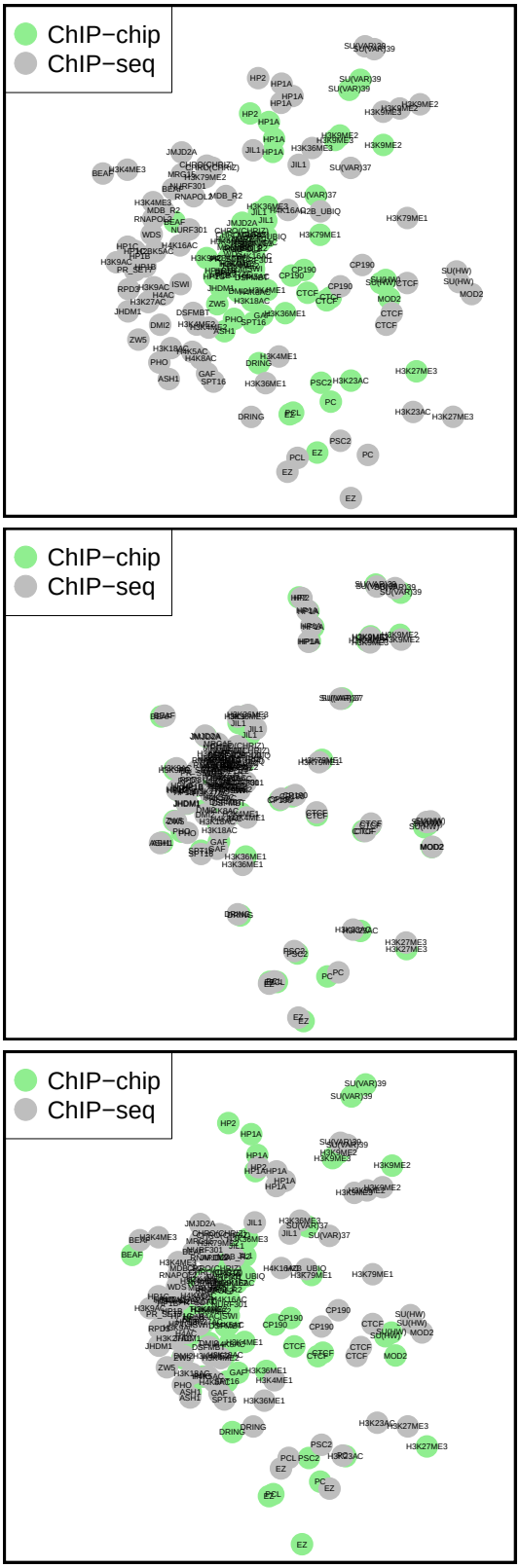
In recognition that such biases may be present whenever combining data from multiple sources, we proposed Procrustes and Peak Width Adjustment as bias-removing methods. Procrustes analysis (Kendall, 1989) combines two (or more) sets of points $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_m)$ by estimating a location, scale and rotation transformation. The shift is estimated as the difference between the mean coordinates for $\mathbf{x}$ and $\mathbf{y}$, whereas the scale and rotation are determined

from their covariance matrices. Procrustes is a general adjustment in the sense that it does not require careful consideration of what caused the systematic differences between sources, but it requires a minimal number of common points between $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and $(\mathbf{y}_1, \ldots, \mathbf{y}_m)$. Specifically, estimating the shift requires at least one common point, while estimating the scale and rotation requires two. In practice, we observed that a reliable match usually requires ≥ 3 common samples. Figure 2 shows how a Procrustes adjustment effectively matched the ChIP-seq samples with their ChIP-chip counterparts, as discussed in the manuscript.

Typically Procrustes adjustment is always applicable for chroGPS-genes maps, as each point is defined by a gene, *i.e.* there are $> 10,000$ common points for the matching. However, for chroGPS-factors maps there may not be enough (or any) common points to use Procrustes. To consider this possibility, we proposed Peak Width Adjustment (PWA) as an alternative method. We found that an important source of systematic differences between technologies arises from the difference in peak widths, *i.e.* their inherent resolution. For instance, ChIP-seq experiments typically produce narrower peaks than matching ChIP-chip experiments. PWA selects the data source i with widest genomic intervals and adjusts interval widths from all other data sources $j \neq i$ so that their mean and variance are equal to that in i. Let (m_{ik}, v_{ik}) be the mean and variance (respectively) of interval log-widths for factor $k = 1, \ldots, n$ in source i, and (m_{jk}, v_{jk}) be those for factor $k = 1, \ldots, m$ in source j. We compute the grand mean $\bar{m}_i = \frac{1}{n}\sum_{k=1}^{n} m_{ik}$ and variance $\bar{v}_i = \frac{1}{n}\sum_{k=1}^{n} v_{ik}$ for source i, and similarly $(\bar{m}_j, \bar{v}_j)$ for source j. Since source i has wider intervals $(\bar{m}_i > \bar{m}_j)$, the log-width w of all intervals in source j is adjusted by computing $\tilde{w} = \bar{m}_i + (w - \bar{m}_j) * \sqrt{\frac{\bar{v}_i}{\bar{v}_j}}$.

We note that PWA assumes that differences in peak width are due to a technical bias, but it is also possible that factors studied in source j truly have a fuzzier genomic location. To address this issue, whenever sources i and j share ≥ 3 factors we only use the common factors to calibrate the adjustment, *i.e.* to estimate $(\bar{m}_i, \bar{v}_i, \bar{m}_j, \bar{v}_j)$. Supplementary Figure 6 shows the peak width distribution for S2 ChIP-chip and ChIP-seq factors. Before adjustment ChIP-chip binding sites are substantially wider than their matching ChIP-seq counterparts, consistently with the lower resolution of these microarray-based experiments. After PWA the peak width distributions have been effectively matched, while preserving the relative width between factors within each technology.

An important limitation for PWA is that it only adjusts the peak-calling resolution, thus ignoring other potential biases (*e.g.* cross-hibridization, sequencing affinity, batch effects). In general we recommend Procrustes for integrating heterogeneous data. However, for the



Supplementary Figure 7. Raw, PWA and Procrustes adjusted chroGPS-factors with ChIP-chip and simulated ChIP-seq data

specific case of combining ChIP-chip and ChIP-seq data the dominating bias is often peak-calling resolution, and hence PWA is a reasonable alternative. In fact, when the only source of bias is peak-calling resolution PWA can provide improved results relative to Procrustes, as we now illustrate with a simulation experiment.

We selected the 76 ChIP-chip studies in S2, and generated 76 synthetic ChIP-seq counterparts. First, we selected each ChIP-chip binding site for inclusion in the simulated ChIP-seq profile with probability 0.9. Second, each interval width was reduced by a factor $r = l_s/l_c$, where l_s and l_c were drawn at random from the observed ChIP-chip and ChIP-seq binding site lengths. Figure 7(top) shows the chroGPS-factors map obtained by merging the data with no adjustment. ChIP-seq factors are systematically shifted and form an outer layer around their ChIP-seq counterparts, similar to what we observed with experimental ChIP-seq data (Figure 5, right). PWA matched points almost perfectly (Figure 7, middle). Procrustes effectively removed the overall shift and scale changes, but the factor matching is less satisfactory than for PWA (Figure 7, bottom). As a final remark, the discussed adjustments target systematic biases between data sources rather than sample-specific biases, *e.g.* chromatin preparation, antibody used for immuno-precipitation. The latter are present even in samples from a single source, hence we view them as a source of (technical) variability rather than systematic biases between sources. Whenever replicated experiments for a single factor are available, this technical variability can be assessed by checking their positions in the map.

5.2. **Conservation assessment.** The first step in assessing conservation across several conditions or backgrounds is to generate a joint map. The joint map is produced by computing distances between points from all sources into a single matrix, and representing them in a low-dimensional space via MDS. For instance, for chroGPS-factors we produced a joint *Drosophila* S2-BG3 map. Because no adjustment strategy was used in this first map, S2 *vs.* BG3 distances are interpretable. An S2 factor appearing nearby its BG3 counterpart indicates that they locate in similar genomic regions, *i.e.* that their genomic location is conserved. We found that most S2-BG3 replicated factors were close in the chroGPS-factors map, although their average S2-BG3 iOverlap was a moderate 60%. This strategy is also applicable to chroGPS-genes maps, so that for each gene we have two separate points characterizing its S2 and BG3 factors. One could then compare the two locations for a given gene (or gene set), nearby positions indicating a conservation of the gene(set) epigenetic profile.

The strategy above assesses *point-wise* conservation in the map, *e.g.* genomic locations of a given factor in chroGPS-factors or the epigenetic profile of a gene in chroGPS-genes. Often it is also of interest to

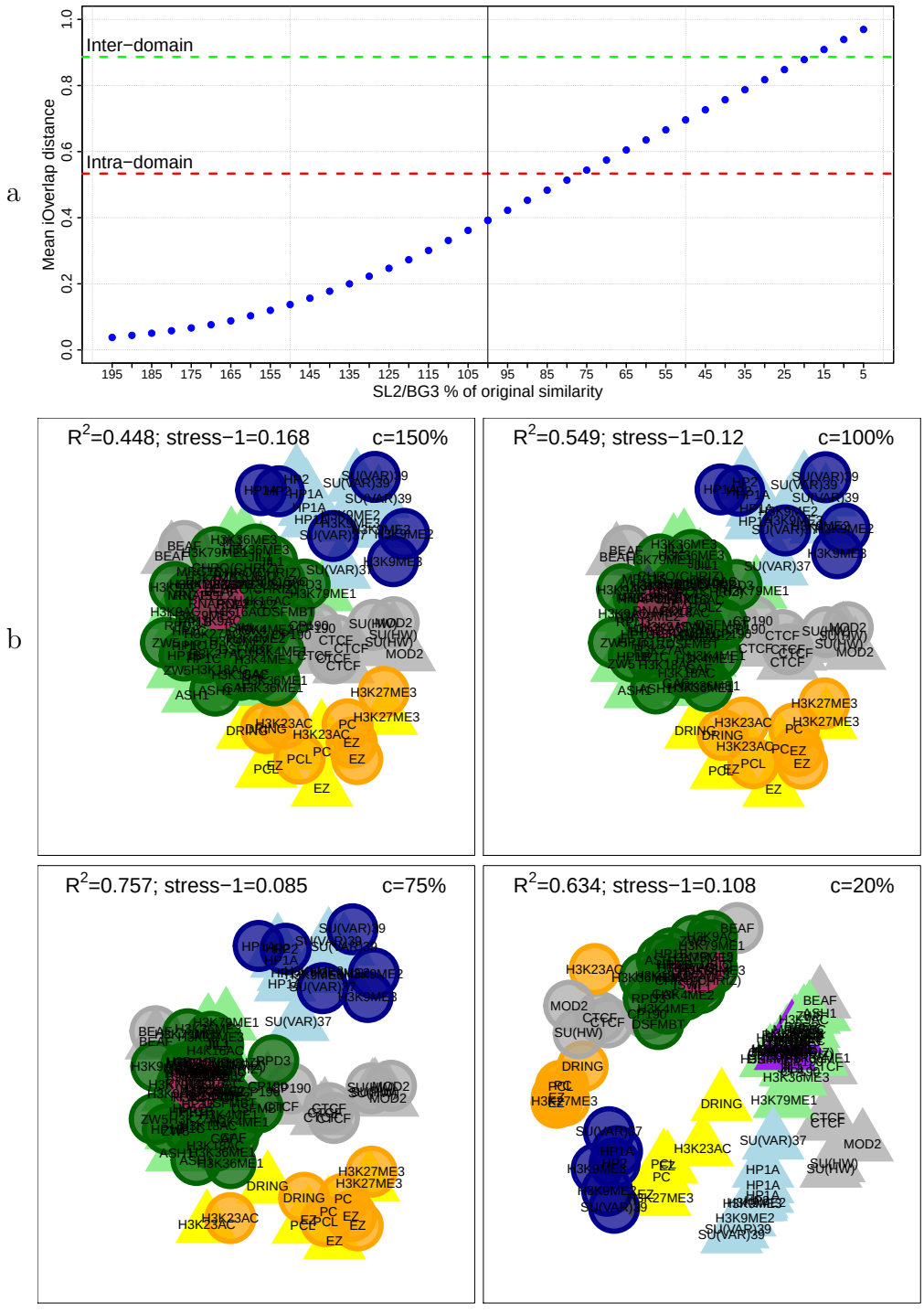
check whether the *relationship between a set of points* is conserved, even when their individual locations are not. For instance, suppose that in the chroGPS-genes map S2 factors located far from their BG3 counterparts, but that the within S2 distances were similar to the within BG3 distances. Such a finding would suggest that, while the genomic location of factors is not conserved across S2 and BG3, their functional interplay is conserved.

We investigated this use by producing chroGPS-factors where distances within S2 and BG3 were preserved, while distances between S2 and BG3 were artificially modified. Specifically, we selected factors replicated in S2 and BG3 and altered their S2-BG3 iOverlap by a fixed proportion $c = \{0.05, 0.1, \ldots, 1.95\}$, *i.e.* defined new distances $\tilde{d}^o_{ij} = 1 - cs^o_{ij}$ when (i, j) were factors both in S2-BG3 and maintained $d^o_{ij} = 1 - s^o_{ij}$ otherwise. Supplementary Figure 8a shows that for $c = 1$ (unaltered distances), the average distance between replicates is 0.39, which is substantially lower than the average intra-domain (0.53) and inter-domain distances (0.89). Accordingly, the map locates S2 factors close to their BG3 counterparts (Supplementary Figure 8b). For $c = 1.5$ (S2-BG3 iOverlap is 150% its original value) domains remain clearly separated and S2-BG3 samples appear slightly closer, as expected. For $c = 0.75$ (S2-BG3 iOverlap is 75% its original value) distances between replicates become comparable to the average intra-domain distance. The general configuration of the map, which is mainly determined by inter-domain distances, is maintained. Interestingly, at $c = 0.20$ distances between replicates have grown comparable to inter-domain distances. The corresponding map adequately reflected the distances between domains and between replicates, while preserving the general domain structure within each cellular background. That is, the map correctly detected a situation where individual factors were differentially located between backgrounds but their functional interplays were conserved.

6. BoostMDS for high-dimensional MDS plots

Producing MDS plots with thousands of points, such as in chroGPS-genes, is a high-dimensional problem that poses important challenges (Buja et al., 2008; Andrecut, 2009). On one hand, classical MDS may fail to converge to a global optimum, resulting in a poor representation. Second, the required computation time may be prohibitive. Supplementary Figure 4 shows that classical MDS achieved relatively low R^2 and stress-1 for the four proposed chroGPS-genes metrics, and required substantial computational time (around 50 minutes) isoMDS improved both R^2 and stress-1, but its computational time was prohibitive for most metrics.

To overcome these limitations, we propose a novel two-step procedure, which we denominate BoostMDS. In the first step, we obtain an



Supplementary Figure 8. Conservation of S2-BG3 binding sites. (a) iOverlap distance between S2-BG3 replicates *vs.* similarity factor; (b) maps for S2-BG3 similarity 150%,100%,70%,20% that in the observed data

initial solution using a variant of the FastMDS method (Yang et al., 2006). We split the problem of representing an $n \times n$ distance matrix into k smaller $m \times m$ successive sub-matrices ($m << n$) in which $j < m$ terms are common between each pair of successive sub-matrices. Since $m << n$, the k independent sub-problems are computationally tractable and can be analyzed in parallel using any MDS algorithm, *e.g.* classical scaling or isoMDS. For the *Drosophila* chroGPS-genes map we used m =2932 and j =56 (*i.e.* 26% and 0.05% of the 11,277 original elements), resulting in k =4 sub-matrices. The k solutions are stitched successively into an overall map using Procrustes adjustment estimated from their j common points. Procrustes is a particular case of the general Affine transformation method used in the FastMDS algorithm. We favor Procrustes due to its preserving the original shape of each sub-map, as only translation, rotation and scaling are allowed, and its straight-forward application to 3D coordinates (Solomon and Breckon (2011), Chapter 7). In contrast, affine transformations use additional uniform scaling and vertical/horizontal shear that can alter angles and relative distances, and application to 3 or more dimensions can be cumbersome. While computationally convenient, the results from this first step depend on an arbitrary split of the distance matrix and there is no guarantee of achieving an optimum in terms of R^2 or stress.

The second step in BoostMDS addresses these issues by formally maximizing the overall R^2. We extended the gradient descent algorithm proposed by Strickert et al. (2007) to automatically set the step size using a combined grid and quadratic search. As R^2 is invariant to location and scale changes, the algorithm is applied to standardized coordinates (z-scores) so that the initial step size is well-calibrated. Let $\mathbf{x}_i^{(k)}$ be the position of point i at iteration k, and $\mathbf{g}_i^{(k)}$ be the gradient evaluated at $\mathbf{x}_i^{(k)}$. At iteration k, we update $\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)} + \lambda \mathbf{g}^{(k-1)}$, where λ is the step size and is set following Algorithm 2.

Algorithm 2. BoostMDS step size determination

(1) Evaluate R^2 for a grid of 5 values $\lambda = (0.1r, \ldots, 0.5r)$, where r is a step size parameter initialized at 0.01. Set $\hat{\lambda}$ to the value achieving largest R^2.
(2) Fit a quadratic regression of R^2 *vs.* λ via least squares, and find its predicted arg-maximum λ^*. If the corresponding R^2 is larger than that for $\hat{\lambda}$, update $\hat{\lambda} = \lambda^*$.
(3) Update $r = \hat{\lambda}$.

That is, at each iteration we determine the optimal value of λ by combining a grid search and a quadratic fit. Only moves increasing R^2 are accepted, and the search stops when the increase in R^2 is negligible (by default, $< 10^{-3}$) or the maximum number of iterations is reached (by default, 50).

We assessed the performance algorithm in the *Drosophila melanogaster* S2 chroGPS-genes data. For stitching sub-maps we used classical MDS, but isoMDS resulted in very similar solutions. Supplementary Figure 4 shows that BoostMDS required substantially less computational time (*e.g.* from 50 minutes to 10 minutes for the Overlap metric) and provided better R^2 and stress-1 than classical MDS and isoMDS. We remark that classical MDS seeks to minimize the stress-1 (2), but it failed to do so as BoostMDS provided a better solution in terms of stress-1. These results illustrate that the classical scaling algorithm may fail to converge in high dimensions.

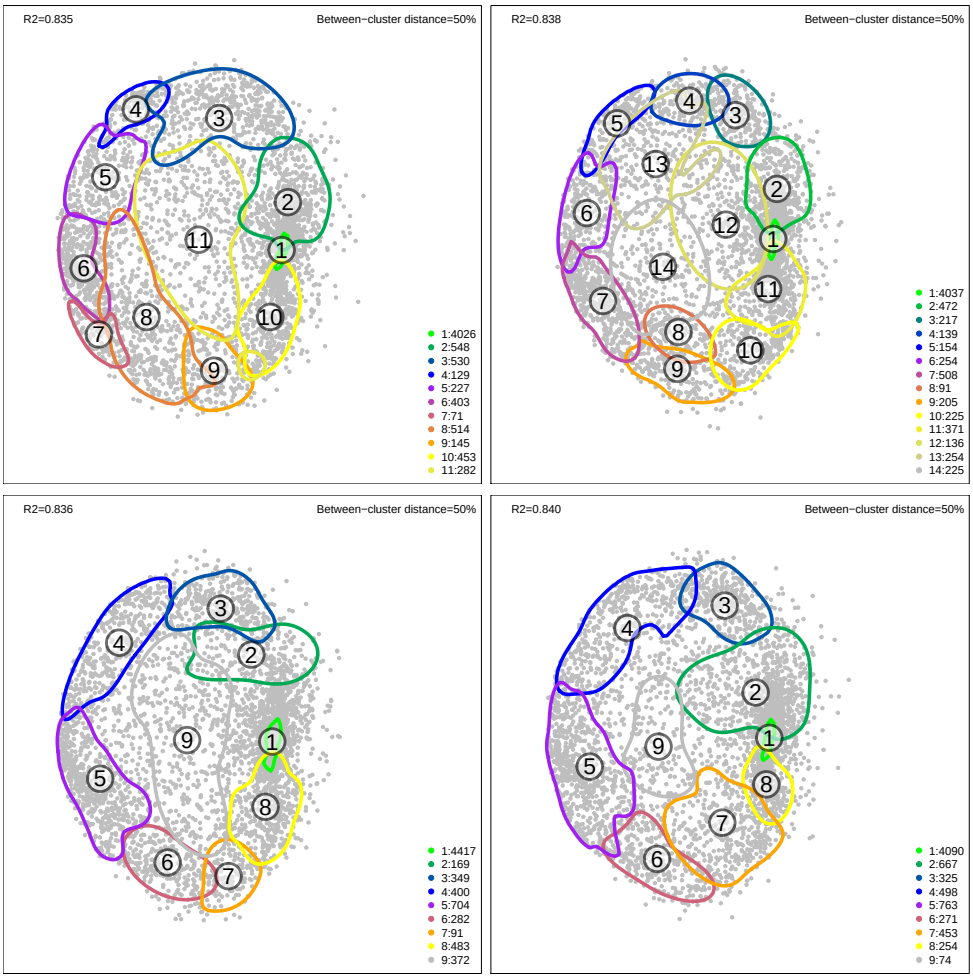
Supplementary Figure 4 shows chroGPS-genes maps of *Drosophila* S2 cells for the four proposed metrics and classical MDS, isoMDS and BoostMDS. By default we recommend using iTanimoto to measure similarities and BoostMDS to represent them, as this strategy generates accurate maps in a reasonable computational time. In order to provide a flexible approach our software also implements Weighted Tanimoto, average overlap and chi-square metrics, as well as multiple MDS algorithms.

7. Clustering analysis and reproducibility in the map

Interpreting a dense maps containing thousands of points, such as chroGPS-genes maps, can be a challenging task. We found that representing the results of simple clustering analyses can greatly facilitate this endeavour. The strategy is to define clusters based on the exact distances (*i.e.* without resorting to the map) and then projecting those clusters in the map. While in principle any user-defined clustering can be used, in our illustrations we used hierarchical clustering with average linkage.

An important question in such an analysis is to decide on the adequate number of clusters. While the choice should be supported by biological and context-specific considerations, we now provide some guidelines. A popular approach is to assess cluster reproducibility via Bootstrap resampling. Importantly, clusters obtained from the exact distance matrix being reproducible is a necessary condition, but not sufficient for clusters being reproducible in the map (due to showing approximated rather than exact distances). That is, even if a Boostrap analysis (based on the exact distances) indicates a cluster to be reproducible, we might not be able to represent it reliably in our low-dimensional space. Our approach is to first merge clusters in the map to guarantee that they are clearly distinguishable in the map (see below for details), and then applying the same procedure to bootstrapped data to check the cluster robustness.

Supplementary Figure 9 shows the map obtained after merging clusters in four bootstrapped datasets. Supplementary Figure 10 (panel

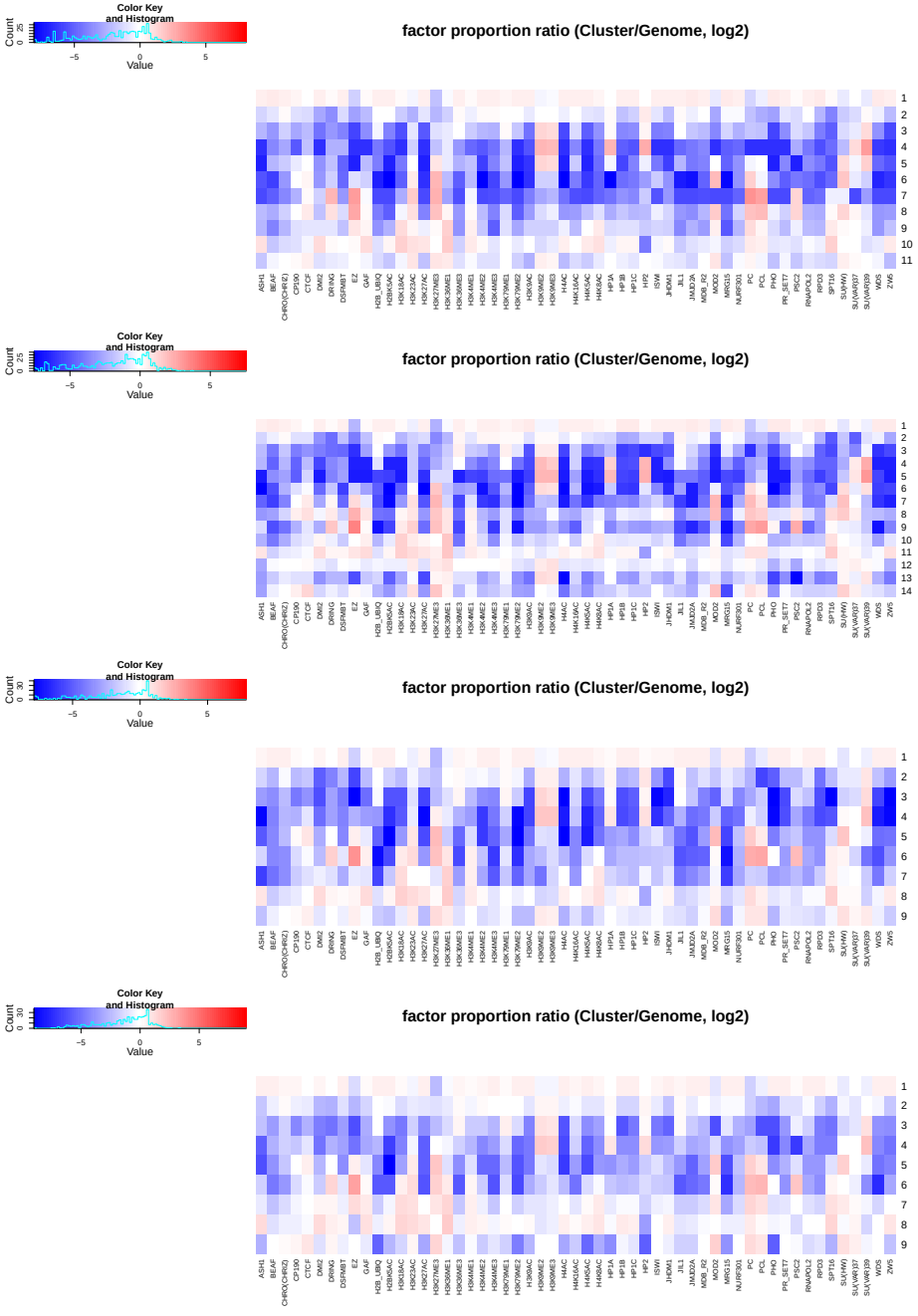


Continued on next page

c, right) shows the same map obtained from the original (*i.e.* non-bootstrapped) data. The original clusters are fairly robust, either re-appearing in the bootstrap maps or being split/merged with another cluster. More importantly, the biological interpretation of map regions provided by the clusters is essentially the same.

We now explain the steps used to merge clusters based on their positions in the map. Intuitively, clusters are well separated when most observations in the map can be unequivocally assigned to a single cluster. More precisely, suppose that a given between-cluster distance threshold t defines K clusters. The probability that element $i = 1, \ldots, n$ belongs to cluster k given its coordinates $\mathbf{x}_i$ is a direct measure of classification power, and is given by Bayes theorem as

$$P_i(k \mid \mathbf{x}_i) = \frac{f_k(\mathbf{x}_i)P(k)}{\sum_{j=1}^{K} f_j(\mathbf{x}_i)P(j)}, \tag{8}$$



SUPPLEMENTARY FIGURE 9. Bootstrap samples with chroGPS-genes maps and clusters (previous page) and epigenetic profiles (above) for between-cluster distance threshold of 0.5. Contours indicate 0.75 probability region for each cluster. Bootstrap clusters are numbered and colored in the same anticlockwise sense used for original clusters as seen in Supplementary Figure 10, row c, right panel.

where $f_k(\mathbf{x}_i)$ is the probability density function for points in cluster k in the map and $P(k)$ is the (known) proportion of points in cluster k. Plugging in $k = C_i$ in (8), where C_i is the cluster that the clustering algorithm assigned to i, measures the certainty in cluster assignment for point i. We obtain the expected correct classification rate (CCR) by averaging over all points

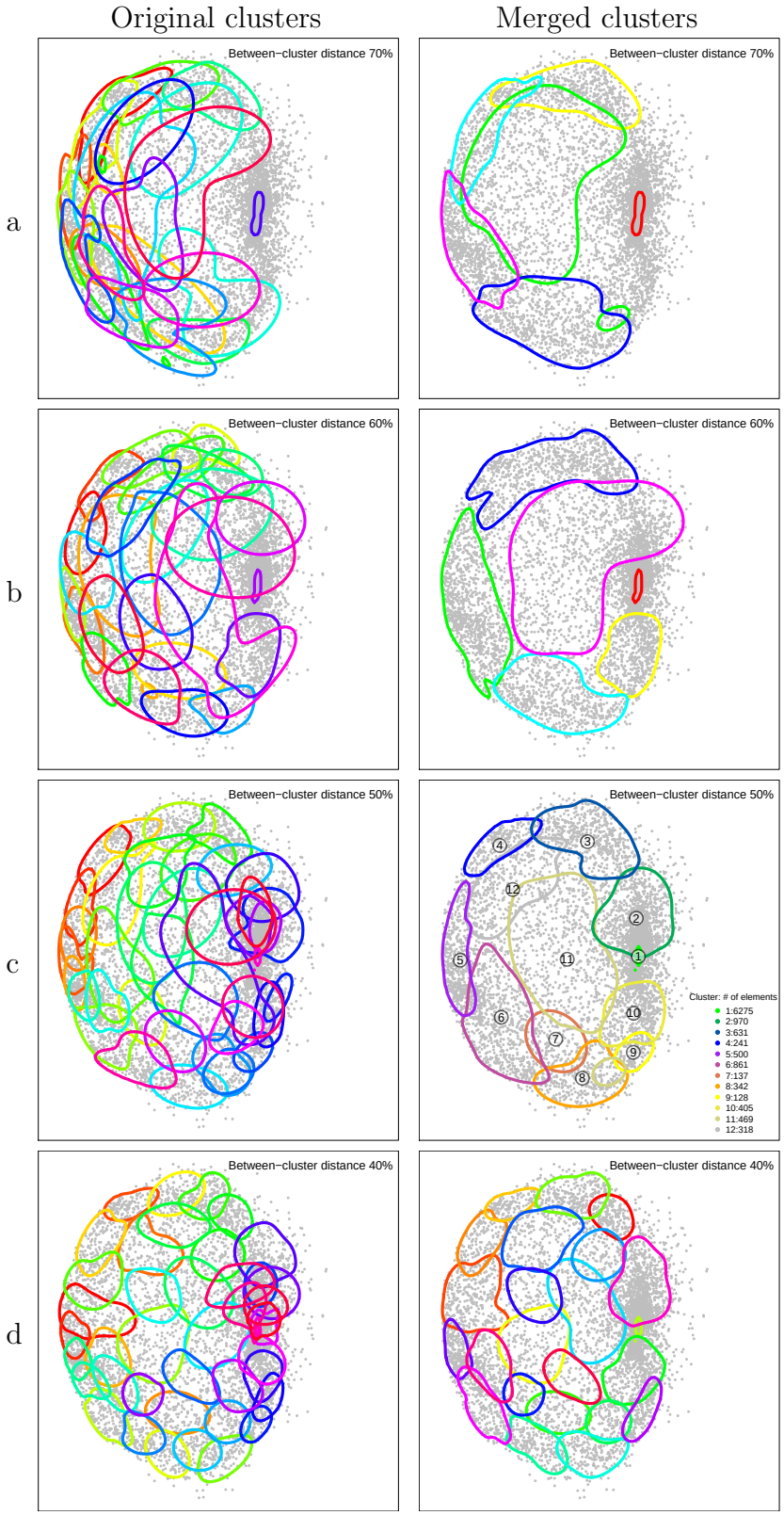
$$\text{CCR} = \frac{1}{n}\sum_{i=1}^{n} P_i(C_i \mid \mathbf{x}_i) \tag{9}$$

and use it as an overall measure of cluster separation. A low CCR value indicates that clusters are not well separated in the map, *i.e.* they either were not separated in their original high-dimensional space or the approximated distances did not accurately represent the cluster in the map. The lowest possible value for CCR is $1/K$, where K is the number of clusters, and is obtained when clusters $k = 1, \ldots, K$ overlap completely (f_k is constant) and have the same proportion of points ($P(k)$ is constant). Conversely, the highest possible value is CCR=1, which indicates perfect separation.

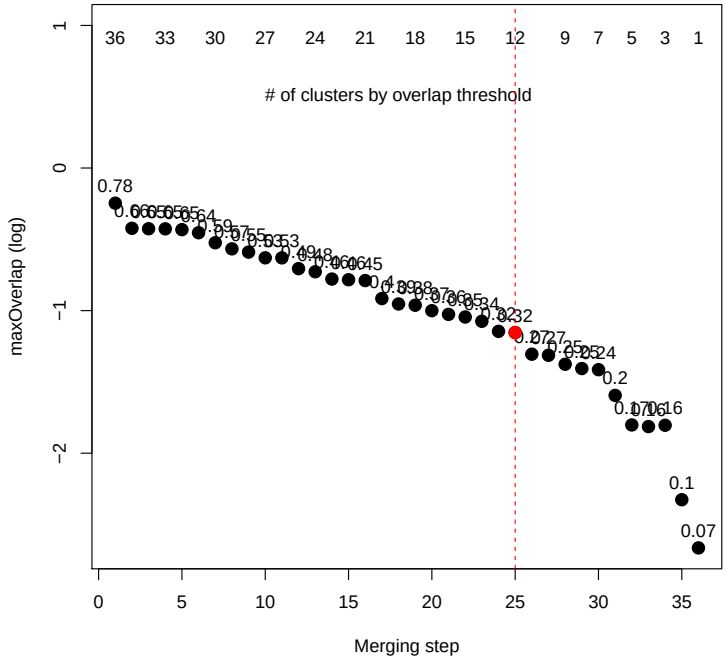
Cluster-specific densities f_k in (8) are unknown and must be estimated from the data non-parametrically, which is problematic for clusters with a small number of points. To address this issue we start by pre-merging small clusters (by default, with < 70 elements) with their closest cluster in terms of distance between centroids. The closest cluster can be of any size, and pre-merging continues until no small clusters remain. We then estimate f_k non-parametrically separately for each cluster using Dirichlet Process mixtures of normals (Escobar and West, 1995), as implemented in function `DPdensity` in the R package DPpackage (Jara et al., 2011). Cluster pre-merging and CCR calculation is implemented in function `clusGPS` in our chroGPS package (for further details see the package documentation). The estimated f_k also allow plotting probability regions for each cluster in the map.

We illustrate our approach in the *Drosophila* S2 chroGPS-genes map, obtained with Tanimoto distances (Section 4.1) and BoostMDS (Section 6). Supplementary Figure 10 (left) shows maps for different between-cluster distance thresholds $t = 0.7, 0.6, 0.5, 0.4$ (*i.e.* average Tanimoto similarity between clusters $\leq 0.3, 0.4, 0.5, 0.6$), and 0.75 probability contours. The number of clusters is large and they are not well separated, complicating map interpretation. Accordingly, the CCR ranges from 0.72 for $t = 0.7$ to 0.46 for $t = 0.4$ (Supplementary Figure 12, left), indicating limited separation.

Because our focus is in obtaining clusters that help interpret distinct regions in the map, whenever the CCR is low we continue merging clusters until good separation is achieved. The pre-merging step provided large enough clusters to estimate f_k, hence we now merge them based



Supplementary Figure 10. chroGPS-genes clusters for between-cluster distance thresholds 0.7 (a), 0.6 (b), 0.5 (c), 0.4 (d). Contours indicate 0.75 probability region for each cluster

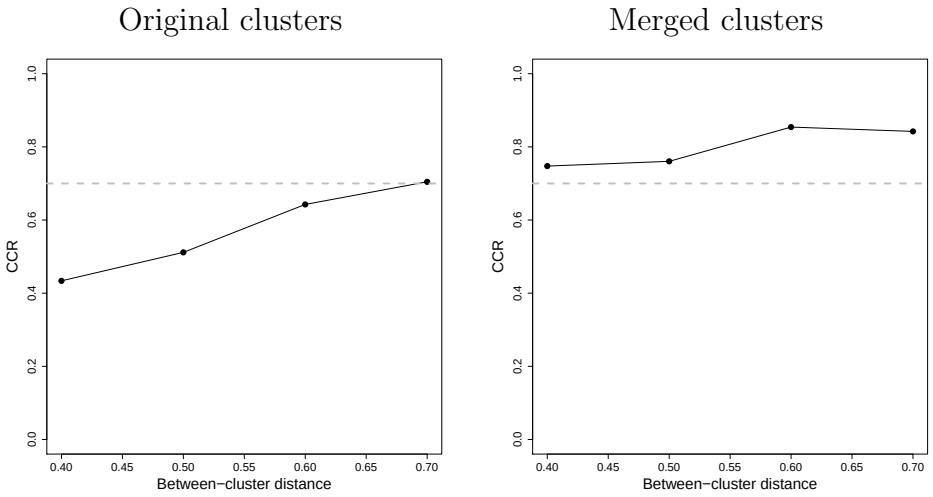


SUPPLEMENTARY FIGURE 11. Probabilistic overlap as chroGPS-genes clusters are merged (axis in log-scale, point labels indicate original value). Red line: change-point detected by cpt.mean

on their probabilistic overlap. More precisely, we define the probabilistic overlap between clusters (k, k') as $\text{PO}_{kk'} =$

(10)
$$\frac{1}{N_k + N_{k'}} \left(\frac{f_k(\mathbf{x}_i)}{f_k(\mathbf{x}_i) + f_{k'}(\mathbf{x}_i)} \text{I}(C_i = k) + \frac{f_{k'}(\mathbf{x}_i)}{f_k(\mathbf{x}_i) + f_{k'}(\mathbf{x}_i)} \text{I}(C_i = k') \right),$$

where $(N_k, N_{k'})$ are the number of points in clusters (k, k'). $\text{PO}_{kk'}$ is preferable to using centroid distances as in the pre-merging step, since it takes into account the degree of concentration and shape of each cluster. We compute $\text{PO}_{kk'}$ for all pairs of clusters and merge those with highest overlap into a new cluster k^*. Rather than re-estimating its corresponding density from scratch we compute it as $f_{k*}(\cdot) = f_k(\cdot)\frac{P(k)}{P(k)+P(k')} + f_{k'}(\cdot)\frac{P(k')}{P(k)+P(k')}$, which results in an efficient algorithm. Cluster merging can continue until CCR or $\text{PO}_{kk'}$ reach a user-defined threshold. In order to provide an automatic procedure, by default we monitor log-$\text{PO}_{kk'}$ and stop merging when it begins to drop swiftly, *i.e.* a change-point is found with `cpt.mean` in R package changepoint (Killick et al., 2012; Killick and Eckley, 2012). We use

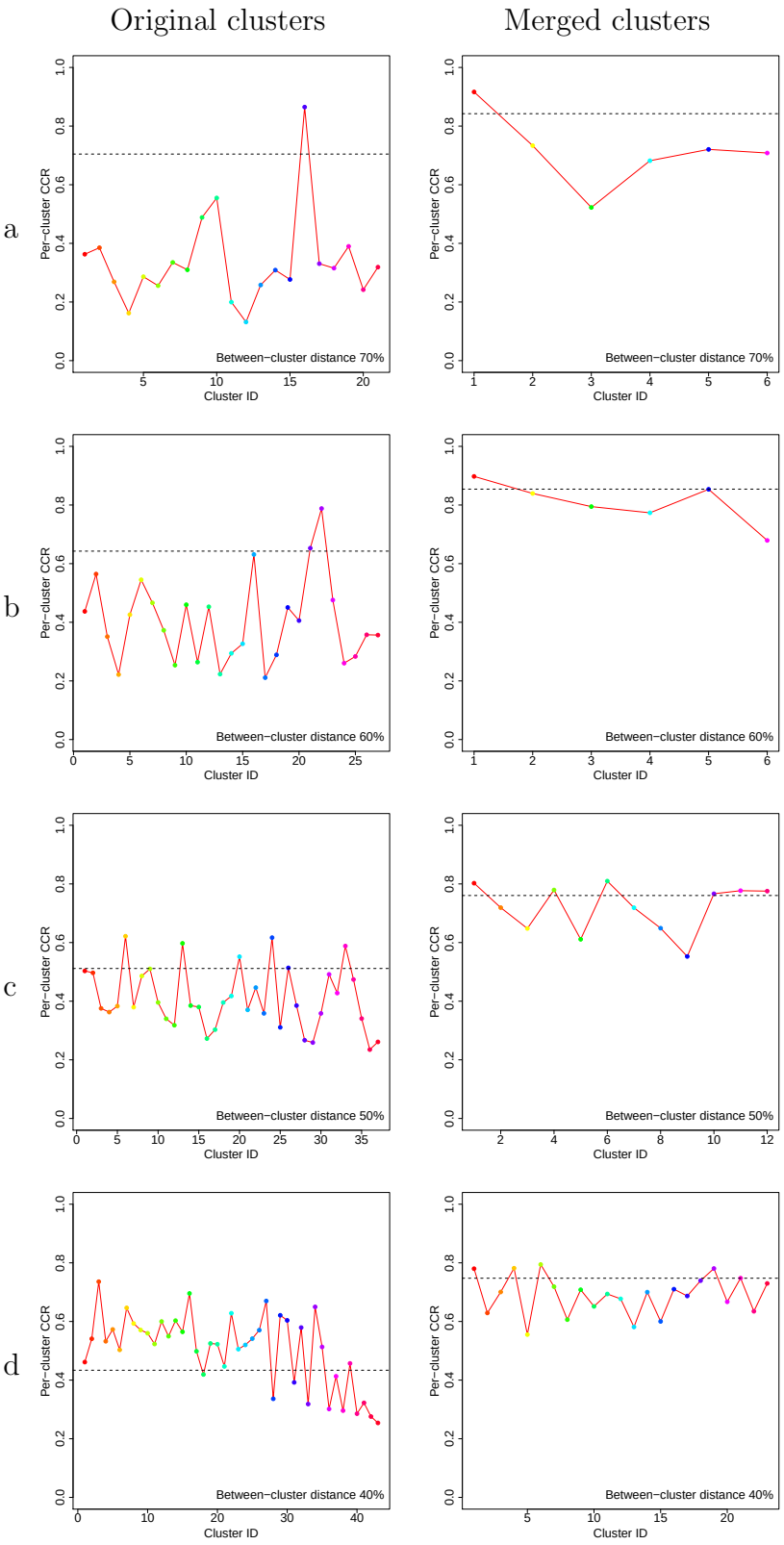


SUPPLEMENTARY FIGURE 12. Posterior expected CCR vs. number of clusters

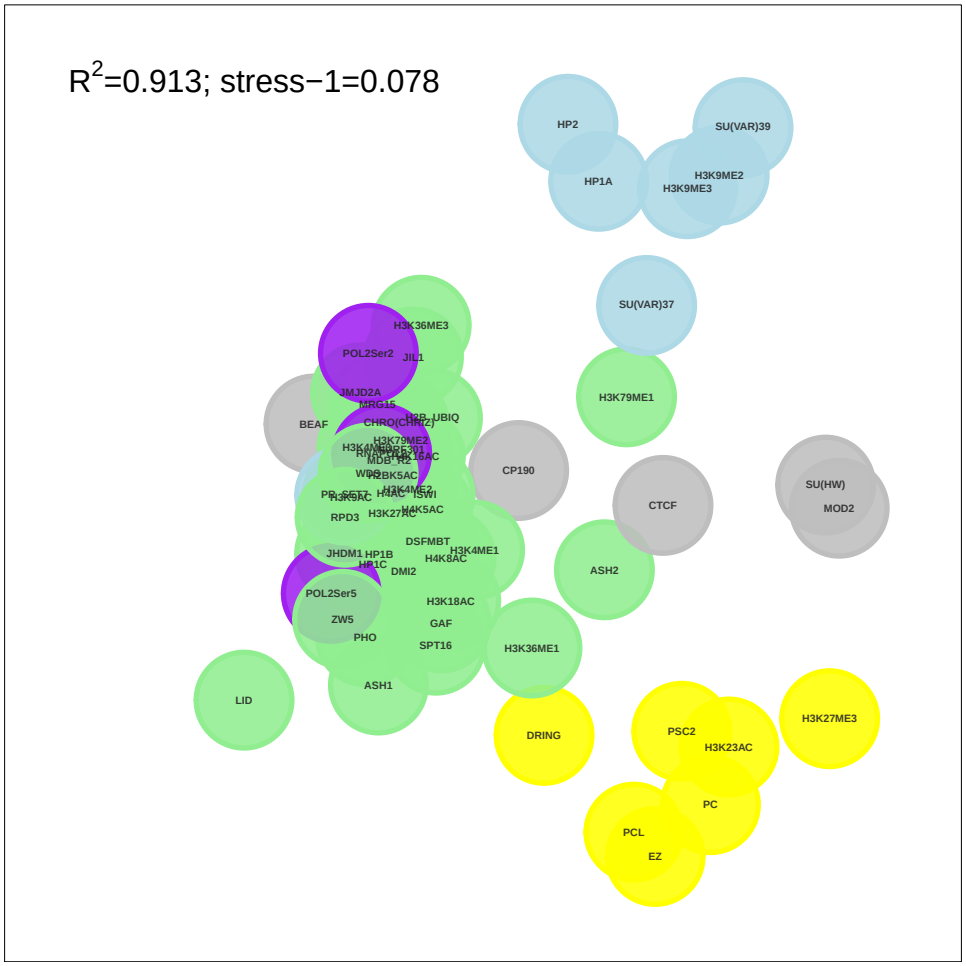
`cpt.mean` with default parameters and `Q=1` change-points. Our strategy, which we implemented in function `mergeClusters`, is similar to that of Baudry et al. (2010) who use information theory metrics instead of $\text{PO}_{kk'}$. Supplementary Figure 11 shows $\text{PO}_{kk'}$ for $t = 0.5$ as clusters are merged and the change-point detected by `cpt.mean`.

Supplementary Figure 10 (right) shows the merged clusters for several between-cluster distance thresholds. Merging produces a smaller number of clusters, which are easier to interpret. As the distance threshold is reduced new clusters appear, providing a higher resolution for interpreting the map. Merging clusters also enforces cluster separation. Supplementary Figure 12 (left) shows the CCR at distance thresholds $t = 0.7, 0.6, 0.5, 0.4$. For the original clusters CCR was generally lower and dropped quickly as t decreased. On the other hand, CCR remained high for merged clusters and was fairly robust to changes in t (Supplementary Figure 12, right). For instance, for $t = 0.7$ (6 clusters) we obtained CCR = 0.87, which agrees with the well-separated clusters observed in Supplementary Figure 10.

Additionally to assessing overall cluster separation, CCR can be separately computed for each cluster (*i.e.* using only points in that cluster in (9)). The cluster-specific CCR indicates how specifically it locates to a region in space. Supplementary Figure 13 shows cluster-specific CCR before and after cluster merging for distance thresholds $t = 0.7, 0.6, 0.5, 0.4$. Before merging CCR is extremely low for some clusters, in accordance with the high degree of overlap observed in Supplementary Figure 10. Merged clusters, on the other hand, show cluster-specific CCR values generally larger than 0.7 or 0.8. That is,



SUPPLEMENTARY FIGURE 13. Cluster-specific CCR vs. number of clusters for between-cluster distance thresholds 0.7 (a), 0.6 (b), 0.5 (c), 0.4 (d). Dotted black line indicates overall CCR



Supplementary Figure 14. Integrated *Drosophila* S2/BG3/WID chroGPS-factors map

after cluster merging each individual cluster was well separated from the rest (in a probabilistic sense).

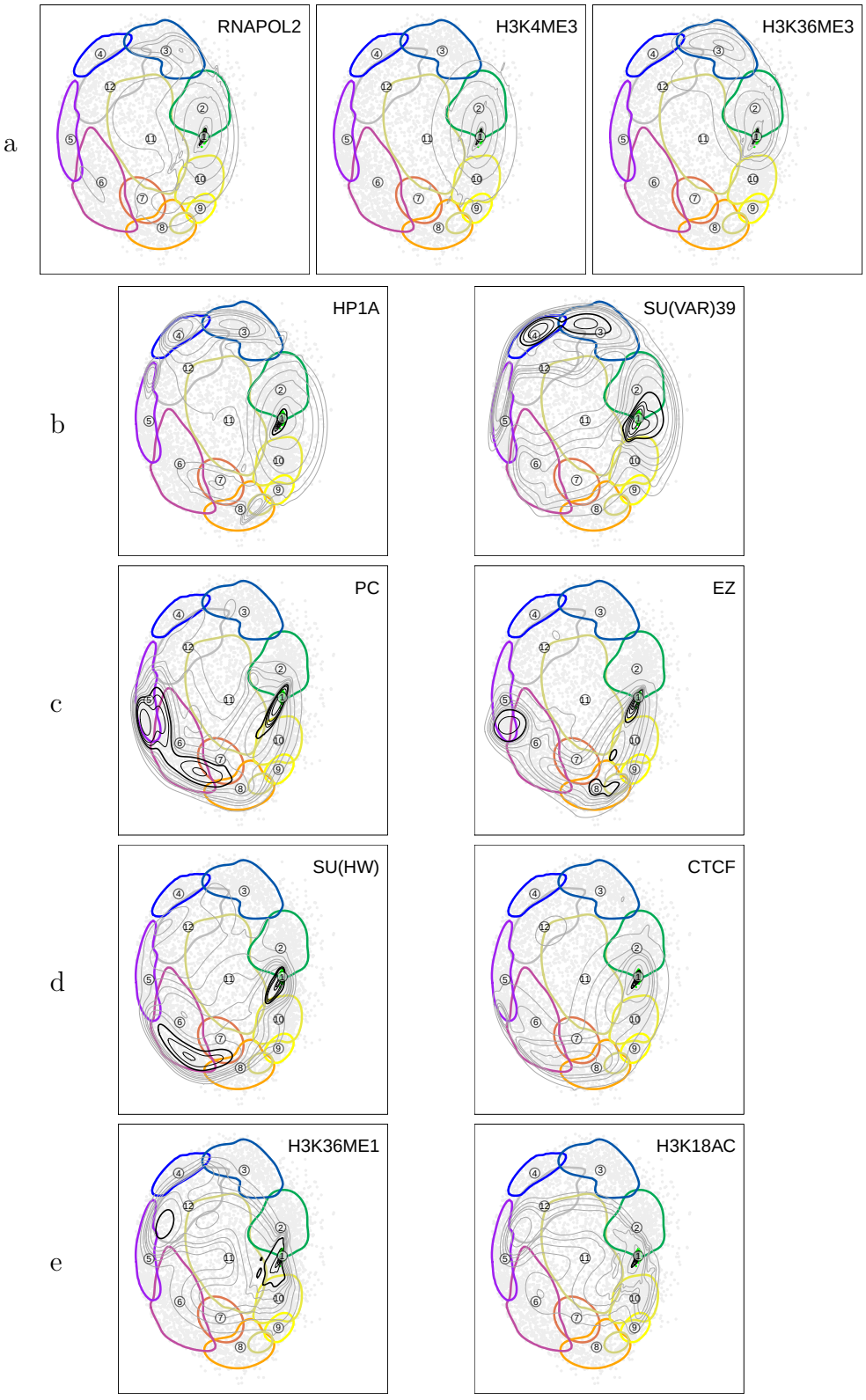
8. Description of the Drosophila chroGPS-factors map

Supplementary Figure 14 shows the *Drosophila* chroGPS-factors map resulting from the integration of 133 genomic profiles obtained from different cell types/tissues (S2, BG3 and wing imaginal disc) by either ChIP-CHIP or ChIP-seq, corresponding to 62 different epigenetic factors. As discussed in the manuscript, the map describes four functional chromatin domains: PC- (yellow) and HP1-dependent silent chromatin (blue), boundaries or insulators (grey) and active chromatin (green) (see also Supplementary Video 1 for an animated view of the 3D map). The PC-silent chromatin domain results from the strong association of six functionally related factors: H3K27me3, the H3K27me3 methyltransferase EZ and the PcG-proteins PC, PSC, PCL and dRING. These

six factors associate very poorly with all the rest, defining a distinct spatial region in the map. Similarly, HP1-silent chromatin domain is defined by the strong association of five factors, H3K9me3, H3K9me2, Su(var)3-9, HP1 and HP2. This association reflects functional co-operation, as Su(var)3-9 is responsible for H3K9me2,3 in heterochromatin and interacts with HP1 that, in addition, binds H3K9me2,3. On the other hand, HP2 interacts and co-localises with HP1 at heterochromatin. A sixth factor, Su(var)3-7, also locates to this domain, indicating that, though more weakly, Su(var)3-7 also associates to HP1-silent chromatin. As a matter of fact, Su(var)3-7 binds to heterochromatin in a HP1-dependent manner but, in addition, it also locates to multiple euchromatic sites. In this respect, it has been shown that Su(var)3-7 affects dosage compensation, suggesting that, a part from heterochromatin formation, Su(var)3-7 plays additional functions in males. Most interestingly, in the map, association of Su(var)3-7 to HP1-silent chromatin is stronger in female BG3 (Figure 3A) than in male S2 cells (Figure 1B and C). Boundary/insulator elements form a third spatial domain. This domain contains three different types of boundary/insulators elements that define distinct sub-domains: BEAF, CTCF and Su(Hw)/Mod. It must be noticed that BEAF is very close to the active-chromatin domain, while both CTCF and Su(Hw)/Mod lay farther away, suggesting that BEAF might be more intimately involved in the regulation of gene expression, as supported by recent genetic data. CP190, which also forms part of this domain, is a common component of all three boundaries/insulators. The active-chromatin domain contains most factors and shows an intricate internal organisation. In particular, the promoter-proximal active RNApol II form (Pol IIoser5), and the elongating RNApol II form (Pol IIoser2) locate in different regions, identifying two distinct subdomains. Notice that most histone acetylations and remodeling factors appear to be associated to promoter-proximal Pol IIoser5, while elongating Pol IIoser2 is associated to H3K36me3, the histone deacetylase RPD3 and the MAPKKK JIL1 that phosphorylates H3S10.

9. Description of the Drosophila S2 cells chroGPS-genes map

Supplementary Figure 15 describes the distribution of selected epigenetic factors on the 12-cluster configuration of the chroGPS-genes map of *Drosophila* S2 cells. Genes in clusters 1 and 2, which are highly expressed, are bound by RNApol II and marked with active histone modifications (H3K4me3 and H3K36me3) (panel a). Notice that genes carrying RNApol II and H3K36me3, but not K3K4me3, are also frequent in cluster 3. Repressive factors show a more complex distribution. For instance, the heterochromatic factors HP1a/Su(var)3-9 are present in the silenced clusters 3 to 5, but they are also frequent in the



SUPPLEMENTARY FIGURE 15. Selected epigenetic factors on the 12-cluster configuration of the S2 chroGPS-genes map in Figure 5A (center). For each factor concentric density contours (10 to 95%) are presented. For each cluster, the 75% density contour is shown.

active chromatin cluster 1 (panel b). Similarly, PC/EZ are frequent in both the silenced clusters 5 to 7 and the active cluster 1 (panel c). Boundary elements, such as Su(Hw) and CTCF, are also enriched in clusters 6 to 8(panel d), suggesting a contribution to the regulation of a subset of PC/EZ-bound genes. Notice that Su(Hw) and CTCF are also present in cluster 1.Cluster 9 is enriched in a group of rather uncharacterized epigenetic factors, such as H3K36me1 and H3K18Ac (panel e), and clusters 10 and 11 appears to correspond to intermediate states Between clusters 9 and 12, being the latter a mixture of Polycomb and Heterochromatin silencing.

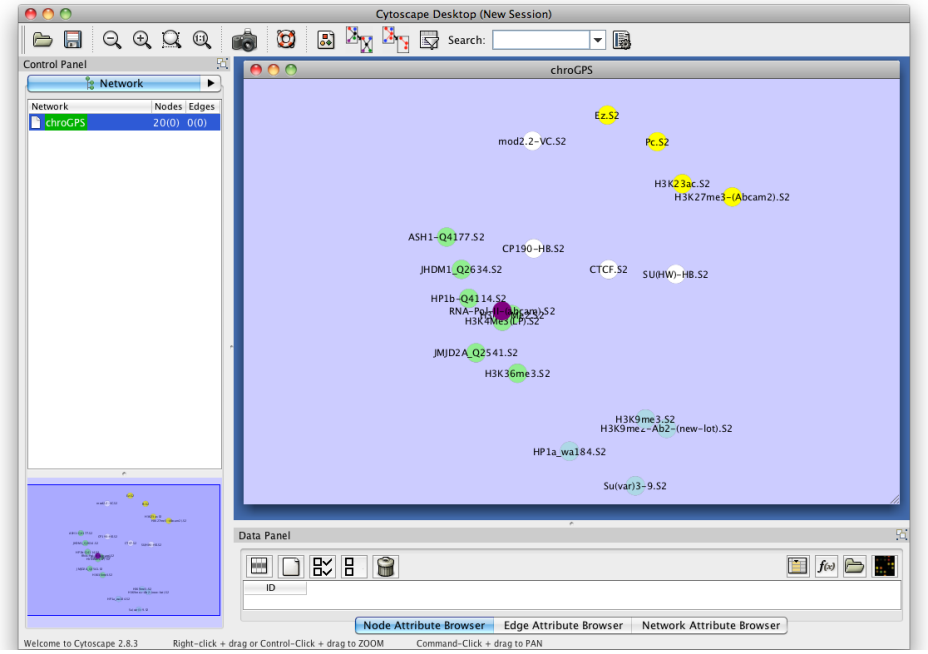
10. Software

Our approach to epigenetic map generation is implemented in the open-source, freely available Bioconductor package `chroGPS` at `http://www.bioconductor.org/packages/2.13/bioc/html/chroGPS.html`. The package contains a dynamic manual (vignette) illustrating the software functionality on a toy subset of the modEncode data, plus a static manual showing the steps required for the full analyses presented in this paper. Both can be accessed either at the URL above or by typing `vignette(package='chroGPS')` at the R prompt.

After 2D and 3D reference maps of epigenetic factors or elements are generated in chroGPS, we offer several alternatives for visualization of such maps. One, as seen on this manuscript, is to simply generate plain figures in PDF or other formats. For improving visualization of 3D maps, chroGPS allows generation of animated 3D videos and also expands visualization and analysis options by exporting chroGPS maps to be used in the Cytoscape software.

10.1. **Visualization: supplementary videos.** The `rgl` package (Adler and Murdoch, 2011) used for visualization of 3D chroGPS maps within R allows creation of animated videos with the desired rotations, points of view and duration. Examples of these videos for chroGPS-factors and chroGPS-genes can be seen in Supplementary Videos 1 and 2 respectively. Both images are in animated GIF format, and can be opened and visualized within any internet browser such as Firefox, Safari, Chrome, etc. Download the file and follow your operating system's instructions to open them with any of those browsers.

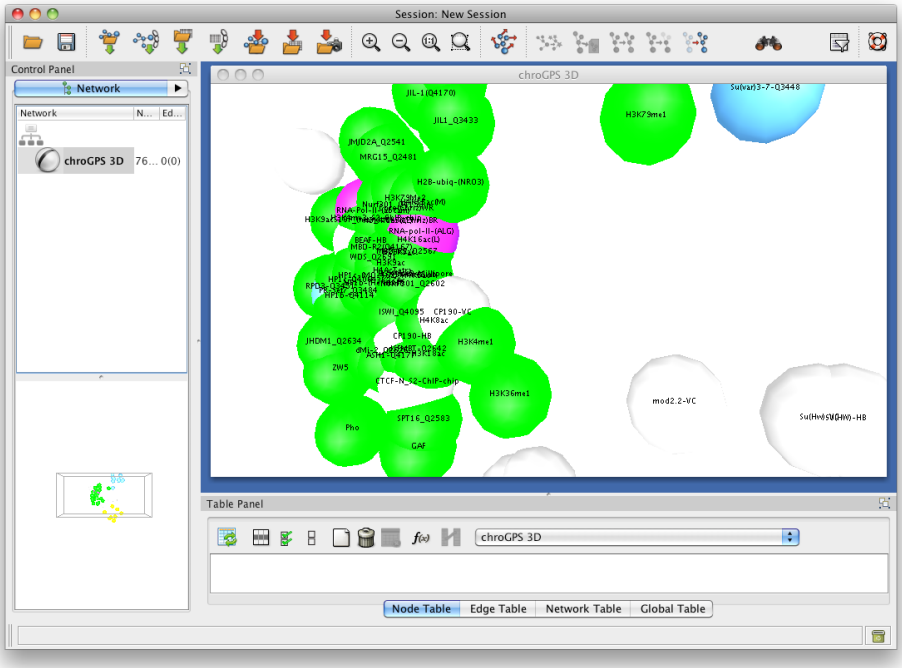
10.2. **Visualization: chroGPS maps to Cytoscape.** Cytoscape `http://www.cytoscape.org` (Shannon et al., 2003) is a widely used software for visualization, annotation and analysis of biological networks. In order to make chroGPS maps easily available to biologists who may not be familiar with the R interface, the function `gps2xgmml` from our `chroGPS` package allows the user to export a chroGPS MDS



SUPPLEMENTARY FIGURE 16. 2D chroGPS-factors map visualized in the Cytoscape software

map as a Cytoscape XGMML network with a simple line of code. Network nodes are identified by their factor (chroGPS-factors) or gene (chroGPS-genes) name, so that importing external information (i.e. expression values) or expanding the original chroGPS object with for instance external regulation networks, Gene Ontology enrichments, etc, becomes natural for Cytoscape users.

The exported network keeps the relative distribution of elements as seen in chroGPS, in order to keep the distances between the original elements intact. For three-dimensional maps Cytoscape 3D Renderer (`http://wiki.cytoscape.org/Cytoscape_3/3D_Renderer`) is required. Supplementary Figures 16 and 17 show respectively 2D and 3D chroGPS-factors map visualized in Cytoscape.



Supplementary Figure 17. 3D chroGPS-factors map visualized using the Cytoscape 3D Renderer

11. Supplementary Tables

ExperimentName	Factor	modEncode ID
ASH1-Q4177.S2	ASH1	modENCODE_2984
BEAF-70.S2	BEAF	modENCODE_922
BEAF-HB.S2	BEAF	modENCODE_274
Chro(Chriz)BR.S2	CHRO(CHRIZ)	modENCODE_278
Chro(Chriz)WR.S2	CHRO(CHRIZ)	modENCODE_279
CP190-HB.S2	CP190	modENCODE_925
CP190-VC.S2	CP190	modENCODE_280
CTCF-N_S2-ChIP-chip	CTCF	modENCODE_913
CTCF-VC.S2	CTCF	modENCODE_283
CTCF.S2	CTCF	modENCODE_3281
dMi-2_Q2626.S2	DMI2	modENCODE_926
dRING-Q3200.S2	DRING	modENCODE_928
dSFMBT-Q2642.S2	DSFMBT	modENCODE_3751
EZ-Q3421.S2	EZ	modENCODE_2988
Ez.S2	EZ	modENCODE_284
GAF.S2	GAF	modENCODE_285
H2B-ubiq-(NRO3).S2	H2B_UBIQ	modENCODE_290
H2BK5ac.S2	H2BK5AC	modENCODE_3283
H3K18ac.S2	H3K18AC	modENCODE_292

Continued on next page

ExperimentName	Factor	modEncode ID
H3K23ac.S2	H3K23AC	modENCODE_294
H3K27Ac.S2	H3K27AC	modENCODE_296
H3K27me3-(Abcam2).S2	H3K27ME3	modENCODE_298
H3K36me1.S2	H3K36ME1	modENCODE_3170
H3K36me3.S2	H3K36ME3	modENCODE_303
H3K4me1.S2	H3K4ME1	modENCODE_304
H3K4me2-Millipore.S2	H3K4ME2	modENCODE_2655
H3K4me2.ab.S2	H3K4ME2	modENCODE_965
H3K4me3_S2-ChIP-chip	H3K4ME3	modENCODE_914
H3K4Me3(LP).S2	H3K4ME3	modENCODE_305
H3K79me1.S2	H3K79ME1	modENCODE_2658
H3K79Me2.S2	H3K79ME2	modENCODE_307
H3K9ac.S2	H3K9AC	modENCODE_309
H3K9acS10P_(new_lot).S2	H3K9AC	modENCODE_2660
H3K9me2-antibody2.S2	H3K9ME2	modENCODE_311
H3K9me2-Ab2-(new-lot).S2	H3K9ME2	modENCODE_3011
H3K9me3.S2	H3K9ME3	modENCODE_313
H4AcTetra.S2	H4AC	modENCODE_201
H4K16ac(L).S2	H4K16AC	modENCODE_319
H4K16ac(M).S2	H4K16AC	modENCODE_320
H4K5ac.S2	H4K5AC	modENCODE_321
H4K8ac.S2	H4K8AC	modENCODE_322
HP1a_552.S2	HP1A	modENCODE_3700
HP1a_wa184.S2	HP1A	modENCODE_2668
HP1a_wa191.S2	HP1A	modENCODE_323
HP1b-(Henikoff).S2	HP1B	modENCODE_941
HP1b-Q4114.S2	HP1B	modENCODE_3020
HP1c-(MO-462).S2	HP1C	modENCODE_943
HP1c-Q4064.S2	HP1C	modENCODE_3291
HP2-(Ab2-90).S2	HP2	modENCODE_944
ISWI_Q4095.S2	ISWI	modENCODE_3032
JHDM1_Q2634.S2	JHDM1	modENCODE_3033
JIL-1(Q4170).S2	JIL1	modENCODE_3038
JIL1_Q3433.S2	JIL1	modENCODE_945
JMJD2A_Q2541.S2	JMJD2A	modENCODE_3784
MBD-R2_Q2567.S2	MDB_R2	modENCODE_946
MBD-R2(Q4167).S2	MDB_R2	modENCODE_3039
mod2.2-VC.S2	MOD2	modENCODE_2674
MRG15_Q2481.S2	MRG15	modENCODE_3047
NURF301_Q2602.S2	NURF301	modENCODE_947
Nurf301_Q4159.S2	NURF301	modENCODE_3048
Pc.S2	PC	modENCODE_326
PCL-Q3412.S2	PCL	modENCODE_3049
Pho.S2	PHO	modENCODE_3894
PR-Set7_Q3484.S2	PR_SET7	modENCODE_3054
Psc.S2	PSC2	modENCODE_3056
RNA-Pol-II-(abcam).S2	RNAPOL2	modENCODE_3295

Continued on next page

ExperimentName	Factor	modEncode ID
RNA-pol-II-(ALG).S2	RNAPOL2	modENCODE_329
RPD3-Q3451.S2	RPD3	modENCODE_3057
SPT16_Q2583.S2	SPT16	modENCODE_3058
SU(HW)-HB.S2	SU(HW)	modENCODE_330
Su(Hw)-VC.S2	SU(HW)	modENCODE_331
Su(var)3-7-Q3448.S2	SU(VAR)37	modENCODE_2672
Su(var)3-9-Q2598.S2	SU(VAR)39	modENCODE_3061
Su(var)3-9.S2	SU(VAR)39	modENCODE_2673
WDS_Q2691.S2	WDS	modENCODE_953
ZW5.S2	ZW5	modENCODE_3804

Supplementary Table 1. modEncode S2 factors

ExperimentName	Factor	modEncode ID
ASH1-Q4177.BG3	ASH1	modENCODE_3279
BEAF-70.BG3	BEAF	modENCODE_921
Chro(Chriz)BR.BG3	CHRO(CHRIZ)	modENCODE_275
CP190-HB.BG3	CP190	modENCODE_924
CTCF-VC.BG3	CTCF	modENCODE_282
CTCF.BG3	CTCF	modENCODE_3280
dRING-Q3200.BG3	DRING	modENCODE_927
dSFMBT-Q2642.BG3	DSFMBT	modENCODE_2986
EZ-Q3421.BG3	EZ	modENCODE_2987
Ez.BG3	EZ	modENCODE_2650
GAF.BG3	GAF	modENCODE_2651
H2B-ubiq-(NRO3).BG3	H2B_UBIQ	modENCODE_288
H3K18ac.BG3	H3K18AC	modENCODE_291
H3K23ac.BG3	H3K23AC	modENCODE_293
H3K27Ac.BG3	H3K27AC	modENCODE_295
H3K27Me3-(Abcam2).BG3	H3K27ME3	modENCODE_297
H3K36me1.BG3	H3K36ME1	modENCODE_299
H3K36me3.BG3	H3K36ME3	modENCODE_301
H3K4me1.BG3	H3K4ME1	modENCODE_2653
H3K4me2-Millipore.BG3	H3K4ME2	modENCODE_2654
H3K4me3.BG3	H3K4ME3	modENCODE_967
H3K79Me1.BG3	H3K79ME1	modENCODE_3005
H3K79Me2.BG3	H3K79ME2	modENCODE_306
H3K79Me3.BG3.Affy_2	H3K79ME3	modENCODE_4197
H3K9acS10P_(new_lot).BG3	H3K9AC	modENCODE_2659
H3K9me2-Ab2-(new-lot).BG3	H3K9ME2	modENCODE_310
H3K9me3-(new-lot).BG3	H3K9ME3	modENCODE_312
H4K16ac(L).BG3	H4K16AC	modENCODE_316
HP1_wa191.BG3	HP1	modENCODE_2666
HP1a_wa184.BG3	HP1A	modENCODE_4126
HP1b-(Henikoff).BG3	HP1B	modENCODE_3016

Continued on next page

ExperimentName	Factor	modEncode ID
HP1c-(MO-462).BG3	HP1C	modENCODE_942
HP2-(Ab2-90).BG3	HP2	modENCODE_3026
ISWI_Q4095.BG3	ISWI	modENCODE_3030
JIL1_Q3433.BG3	JIL1	modENCODE_3035
mod2.2-VC.BG3	MOD2	modENCODE_324
MRG15_Q2481.BG3	MRG15	modENCODE_3045
Pc.BG3	PC	modENCODE_325
PCL-Q3412.BG3	PCL	modENCODE_948
Psc.BG3	PSC	modENCODE_3055
RNA-pol-II-(ALG).BG3	RNAPOL2	modENCODE_950
RPD3-Q3451.BG3	RPD3	modENCODE_4188
SU(HW)-HB.BG3	SU(HW)	modENCODE_951
Su(var)3-7-Q3448.BG3	SU(VAR)37	modENCODE_2671
Su(var)3-9.BG3	SU(VAR)39	modENCODE_952
ZW5.BG3	ZW5	modENCODE_3064

Supplementary Table 2. modEncode BG3 factors

References

D. Adler and D. Murdoch. *rgl: 3D visualization device system (OpenGL)*, 2011. URL `http://CRAN.R-project.org/package=rgl`. R package version 0.92.798.

M. Andrecut. Molecular dynamics multidimensional scaling. *Physics Letters A*, 373:2001,2006, 2009.

J.P. Baudry, A.E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19:332–353, 2010.

J.P. Benzécri. *L'Analyse des Données. Volume II. L'Analyse des Correspondances*. Dunod, Paris, 1973.

Andreas Buja, Deborah F. Swayne, Michael L. Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008.

S.E. Celniker, L.A. Dillon, M.B. Gerstein, K.C. Gunsalus, S. Henikoff, G.H. Karpen, M. Kellis, E.C. Lai, J.D. Lieb, D.M. MacAlpine, G. Micklem, F. Piano, M. Snyder, L. Stein, K.P. White, and R.H. Waterston. modencode consortium. unlocking the secrets of the genome. *Nature*, 459(7249):927–30, 2009.

M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90: 577–588, 1995.

R.C. Gentleman, V.J. Carey, D.M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A.J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J.Y.H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004. URL `http://genomebiology.com/2004/5/10/R80`.

A. Jara, T. Hanson, F. Quintana, P. Mueller, and G. Rosner. Dppackage: Bayesian semi- and nonparametric modeling in r. *Journal of Statistical Software*, 40(5):1–30, 2011.

D.G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99, 1989.

R. Killick and I.A. Eckley. *changepoint: An R package for changepoint analysis*, 2012. R package version 0.6 (`http://CRAN.R-project.org/package=changepoint`).

R. Killick, P. Fearnhead, and I.A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

J.B. Kruskal. Multidimensional scaling by optimizing a goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964a.

J.B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964b.

P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, 13(11):2498–2504, Nov 2003.

R. Sibson. Order invariant methods for data analysis. *Journal of the Royal Statistical Society B*, 34:311–349, 1972.

C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell, Chichester, 2011.

M. Strickert, N. Sreenivasulu, B. Usadel, and U. Seiffert. Correlation-maximizing surrogate gene space for visual mining of gene expression patterns in developing barley endosperm tissue. *BMC Bioinformatics*, 8:165+, 2007.

W.S. Torgerson. Multi-dimensional scaling: I, theory and method. *Psychometrika*, 17:401–419, 1952.

W. N. Venables and B. D. Ripley. *Modern applied statistics with S*. Springer, 4th edition, August 2002. ISBN 0387954570.

T. Yang, J. Liu, L. Mcmillan, and Wang. W. A fast approximation to multidimensional scaling, by. In *Proceedings of the ECCV Workshop on Computation Intensive Methods for Computer Vision (CIMCV*, 2006.

L.J. Zhu, C. Gazin, N.D. Lawson, H. Pagès, S.M. Lin, D.S. Lapointe, and M.R. Green. ChIPpeakAnno: a bioconductor package to annotate chIP-seq and chIP-chip data. *BMC Bioinformatics*, 11:237, 2010.