Interpretable Online Network Dictionary Learning for Inferring Long-Range Chromatin Interactions

Vishal Rana[1], Jianhao Peng[1], Chao Pan[1], Hanbaek Lyu[2], Albert Cheng[3], Minji Kim[4], Olgica Milenkovic[1*]

1 Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign.
2 Department of Mathematics, University of Wisconsin - Madison.
3 School of Biological and Health Systems Engineering, Arizona State University, Phoenix.
4 Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor.

*Corresponding author: milenkov@illinois.edu

Abstract

Dictionary learning (DL), implemented via matrix factorization (MF), is commonly used in computational biology to tackle ubiquitous clustering problems. The method is favored due to its conceptual simplicity and relatively low computational complexity. However, DL algorithms produce results that lack interpretability in terms of real biological data. Additionally, they are not optimized for graph-structured data and hence often fail to handle them in a scalable manner.

In order to address these limitations, we propose a novel DL algorithm called *online convex network dictionary learning* (online cvxNDL). Unlike classical DL algorithms, online cvxNDL is implemented via MF and designed to handle extremely large datasets by virtue of its online nature. Importantly, it enables the interpretation of dictionary elements, which serve as cluster representatives, through convex combinations of real measurements. Moreover, the algorithm can be applied to data with a network structure by incorporating specialized subnetwork sampling techniques.

To demonstrate the utility of our approach, we apply cvxNDL on 3D-genome RNAPII ChIA-Drop data with the goal of identifying important long-range interaction patterns (long-range dictionary elements). ChIA-Drop probes higher-order interactions, and produces data in the form of hypergraphs whose nodes represent genomic fragments. The hyperedges represent observed physical contacts. Our hypergraph model analysis has the objective of creating an interpretable dictionary of long-range interaction patterns that accurately represent global chromatin physical contact maps. Through the use of dictionary information, one can also associate the contact maps with RNA transcripts and infer cellular functions.

To accomplish the task at hand, we focus on RNAPII-enriched ChIA-Drop data from *Drosophila Melanogaster* S2 cell lines. Our results offer two key insights. First, we demonstrate that online cvxNDL retains the accuracy of classical DL (MF) methods while simultaneously ensuring unique interpretability and scalability. Second, we identify distinct collections of proximal and distal interaction patterns involving chromatin elements shared by related processes across different chromosomes, as well as patterns unique to specific chromosomes. To associate the dictionary elements with biological properties of the corresponding chromatin regions, we employ Gene Ontology (GO) enrichment analysis and perform multiple RNA coexpression studies.

Availability and Implementation: The code and test datasets are available at: `https://github.com/rana95vishal/chromatin_DL/`

Author summary

We introduce a novel method for dictionary learning termed *online convex Network Dictionary Learning* (online cvxNDL). The method operates in an online manner and utilizes representative subnetworks of a network dataset as dictionary elements. A key feature of online cvxNDL is its ability to work with graph-structured data and generate dictionary elements that represent convex combinations of real data points, thus ensuring interpretability.

Online cvxNDL is used to investigate long-range chromatin interactions in S2 cell lines of *Drosophila Melanogaster* obtained through RNAPII ChIA-Drop measurements represented as hypergraphs. The results show that dictionary elements can accurately and efficiently reconstruct the original interactions present in the data, even when subjected to convexity constraints. To shed light on the biological relevance of the identified dictionaries, we perform Gene Ontology enrichment and RNA-seq coexpression analyses. These studies uncover multiple long-range interaction patterns that are chromosome-specific. Furthermore, the findings affirm the significance of convex dictionaries in representing TADs cross-validated by imaging methods (such as 3-color FISH (fluorescence in situ hybridization)).

Introduction

Dictionary learning (DL) is a widely used method in learning and computational biology for approximating a matrix through sparse linear combinations of dictionary elements. DL has been used in various applications such as clustering, denoising, data compression, and extracting low-dimensional patterns [1–8]. For example, DL is used to cluster data points since dictionary elements essentially represent centroids of clusters. DL can perform denoising by combining only the highest-score dictionary elements to reconstruct the input; in this case, the low-score dictionary elements reflect the distortion in the data due to noise. DL can also perform efficient data compression by storing only the dictionary elements and associated weights needed for reconstruction. In addition, DL can be used to extract low-dimensional patterns from complex high-dimensional inputs.

However, standard DL methods [9, 10] suffer from interpretability and scalability issues and are primarily applied to *unstructured* data. To address interpretability issues for unstructured data, convex matrix factorization was introduced in [11]. Convex matrix factorization requires that the dictionary elements be convex combinations of real data points, thereby introducing a constraint that adds to the computational complexity of the method. At the same time, to improve scalability, DL and convex DL algorithms can be adapted to online settings [12, 13]. Network DL (NDL), introduced in [14], operates on graph-structured data and samples subnetworks via Markov Chain Monte Carlo (MCMC) methods [14–16] to efficiently and accurately identify a small number of subnetwork dictionary elements that best explain subgraph-level interactions of the entire global network. These dictionary elements learned by the original NDL algorithm only provide 'latent' subgraph structures that are not necessarily associated with specific subgraphs in the network. When applied to gene interaction networks, such latent subnetworks cannot be associated with specific genomic regions or viewed as physical interactions between genomic loci, making the method biologically uninterpretable.

To address the shortcoming of online NDL, we propose online cvxNDL, a novel NDL method that combines the MCMC sampling technique from [14] with convexity constraints on the matrix representation of sampled subnetworks. These constraints are handled through the concept of “dictionary element representatives,” which are essentially adjacency matrices of real subnetworks of the input network. The representatives are used as building blocks of actual dictionary elements. More precisely, dictionary elements are convex combinations of small subsets of representatives. This allows us to map the dictionary element entries to actual genomic regions and view them as real physical interactions. The online learning component is handled via sequential updates of the best choice of representative elements, complementing the approach proposed in [13] for unstructured data. This formulation ensures interpretability of the results and allows for scaling to large datasets.

The utility of online cvxNDL is demonstrated by performing an extensive analysis of 3D chromatin interaction data generated by the RNAPII ChIA-Drop [17] technique. Chromatin 3D structures play a crucial role in gene regulation [18, 19] and have traditionally been measured using “bulk” sequencing methods, such as Hi-C [20] and ChIA-PET [21, 22]. However, due to the proximity ligation step, these methods can only capture pairwise contacts and fail to extract potential multiway interactions that exist in the cell. Further, these methods operate on a population of millions of molecules and therefore only provide information about population averages. ChIA-Drop, by contrast, mitigates these issues by employing droplet-based barcode-linked sequencing to capture multiway chromatin interactions at the single-molecule level, enabling the detection of short- and long-range interactions involving multiple genomic loci. Note that, more specifically, RNAPII ChIA-Drop data elucidates interactions among regulatory elements such as enhancers and promoters, which warrants contrasting/combining it with RNA-seq data.

The cvxNDL method is first tested on synthetic data, and, subsequently, on real-world RNAPII ChIA-Drop data pertaining to chromosomes of *Drosophila Melanogaster* Schneider 2 (S2) phagocytic cell lines[1]. For simplicity, we will henceforth refer to the latter as ChIA-Drop data. Our findings are multi-fold.

First, we provide dictionary elements that can be used to represent chromatin interactions in a succinct and highly accurate manner.

Second, we discover significant differences between the long-range interactions captured by dictionary elements of different chromosomes. These differences can also be summarized via the average distance between interacting genomic loci and the densities of interactions.

Third, we perform Gene Ontology (GO) enrichment analysis to gain insights into the collective functionality of the genomic regions represented by the dictionary elements of different chromosomes. As an example, for chromosomes 2L and 2R, our GO enrichment analysis reveals significant enrichment in several important terms related to reproduction, oocyte differentiation, and embryonic development. Likewise, chromosomes 3L and 3R are enriched in key GO terms associated with blood circulation and response to heat and cold.

Fourth, to further validate the utility of the dictionary elements, we perform an RNA-Seq coexpression analysis using data from independent experiments conducted on *Drosophila Melanogaster* S2 cell lines, available through the NCBI Sequence Read Archive [23]. We show that genes associated with a given dictionary element exhibit high levels of coexpression, as validated on TAD interactions T1-T4 and R1-R4 [17].

[1] Due to the limited number of complete ChIA-Drop datasets, we only report findings for cell-lines also studied in [17].

Notably, a small subset of our dictionary elements is able to accurately represent these TAD regions and their multiway interactions, confirming the capability of our method to effectively capture complex patterns of both short- and long-range interactions. In addition, we map our dictionary elements onto interaction networks, including the STRING protein-protein interaction network [24], as well as large gene expression repositories like FlyMine. We observe closely coordinated coexpression among the identified genes, further supporting the biological relevance of the identified dictionary elements.

With its unique features, our new interpretable method for dictionary learning adds to the growing literature on machine learning approaches that aim to elucidate properties of chromatin interactions [25–28].

Results and Discussion

We first provide an intuitive, high-level overview of the steps of the interpretable dictionary learning method, as illustrated in Figure 1. The figure describes the most important global ideas behind our novel online cvxNDL pipeline. A rigorous mathematical formulation of the problem and relevant analyses are delegated to the Methods Section, while detailed algorithmic methods are available in the Supplement Section 2.

Chromatin interactions are commonly represented as contact maps. A contact map can be viewed as a hypergraph, where nodes represent genomic loci and two or more such nodes are connected through hyperedges to represent experimentally observed multiway chromatin interactions. Since it is challenging to work with hypergraphs directly, the first step is to transform a hypergraph into an ordinary network (graph), which we tacitly assume is connected. For this purpose, we employ *clique expansion* [29, 30], as shown in Figure 1b. Clique expansion converts a hyperedge into a clique (a fully connected network) and therefore preserves all interactions encapsulated by the hyperedge. However, large hyperedges covering roughly 10 or more nodes in the network can introduce distortion by creating new cliques that do not correspond to any multiway interaction, as shown in Figure 1c [31]. The frequency of such large hyperedges and the total number of hyperedges in chromatin interaction data is limited (i.e., the hypergraph is sparse, see Supplement Table 1). This renders the distortion due to the hypergraph-to-network conversion process negligible.

To generate an online sample from the clique-expanded input network, we use a subnetwork sampling procedure shown in Figure 1d. We consider a small template network consisting of a fixed number of nodes and search for induced subnetworks in the input that contain the template network topology. These induced subnetworks can be rigorously characterized via *homomorphisms* and are discussed in detail in the Methods Section. An example of a homomorphism is shown in Figure 1d. Throughout our analysis, we will *exclusively focus on path homomorphisms* because they are most suitable for the biological problem investigated. To generate a sequence of online samples from the input network, we employ MCMC sampling. Given a path sample at discrete time t, the next sample at time $t + 1$ is generated by selecting a new node uniformly at random from the neighborhood of the sample at time t and calculating its probability of acceptance β, explained in the Methods Section. If this new node is accepted, we perform a *directed* random walk starting at the selected node, otherwise, we restart the random walk from the first node of the sample at time t. Note that the input network is undirected while only the sampling method requires a directed walk as the order of the labeled nodes matters. (see Figure 1e).

MCMC sampling is used to generate a sequence of samples to initialize a dictionary with K *dictionary elements,* where K is chosen based on the properties of the dataset.

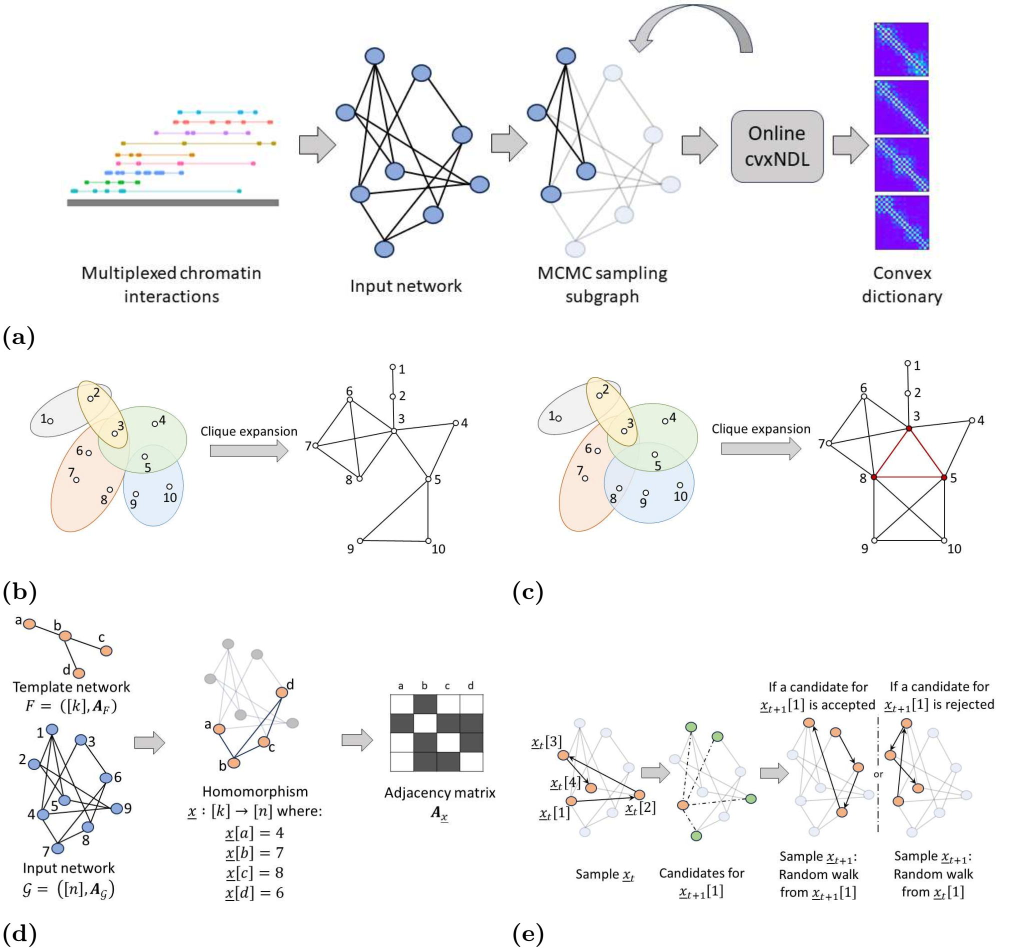

Multiplexed chromatin interactions
Input network
MCMC sampling subgraph
Online cvxNDL
Convex dictionary
(a)
Clique expansion
(b)
Clique expansion
(c)
Template network
$F = ([k], \boldsymbol{A}_F)$
Input network
$\mathcal{G} = ([n], \boldsymbol{A}_\mathcal{G})$
Homomorphism
$\underline{x} : [k] \to [n]$ where:
$\underline{x}[a] = 4$
$\underline{x}[b] = 7$
$\underline{x}[c] = 8$
$\underline{x}[d] = 6$
Adjacency matrix
$\boldsymbol{A}_{\underline{x}}$
(d)
If a candidate for $\underline{x}_{t+1}[1]$ is accepted
If a candidate for $\underline{x}_{t+1}[1]$ is rejected
or
Sample $\underline{x}_t$
Candidates for $\underline{x}_{t+1}[1]$
Sample $\underline{x}_{t+1}$: Random walk from $\underline{x}_{t+1}[1]$
Sample $\underline{x}_{t+1}$: Random walk from $\underline{x}_t[1]$
(e)

Fig 1. (a) Workflow of the dictionary learning method. Multiway (multiplexed) chromatin interactions represented as hyperedges are *clique expanded* into standard networks and combined to create input networks for the algorithm. MCMC subnetwork sampling is then used to generate samples for initialization and online updates during iterative optimization of the objective function, resulting in convex dictionary elements. (b) Illustration of the clique expansion process. Hyperedges are subsets of indexed nodes shaded with the same color. (c) Illustration of clique expansion distortion. There is no hyperedge including nodes 3, 5, and 8 (colored red), and this 3-clique only exists due to shared nodes/edges of "real" hyperedges. Such distortion is negligible when the number of large hyperedges is limited. (d) Subnetwork sampling and the notion of a *motif homomorphism*. These correspond to subnetworks of the input network induced by a fixed number of nodes that contain a template motif topology. The set of homomorphisms $\mathtt{Hom}(F, \mathcal{G})$ for a network $\mathcal{G}$ and the template network F are defined in the Methods Section (Equation 7). Also depicted are an example homomorphism $\underline{x} \in \mathtt{Hom}(F, \mathcal{G})$ and its induced adjacency matrix $\mathbf{A}_{\underline{x}}$ for an input network $\mathcal{G}$ with 9 nodes. The template F is a star network on 4 nodes. In the adjacency matrix, a black field indicates 1, while a white field indicates 0. (e) Workflow of the MCMC sampling algorithm for path homomorphisms. Given a sample $\underline{x}_t$ at time t, obtained via a directed random walk from an initial state in the input network, $\underline{x}_t[1]$, we generate a sample $\underline{x}_{t+1}$ at time $t+1$ by choosing uniformly at random a node v from the neighborhood of $\underline{x}_t[1]$ (marked in green) and calculating a probability of acceptance β. If node v is accepted, we initiate a new directed random walk from v, otherwise, we restart a directed random walk from $\underline{x}_t[1]$ to generate a new sample.

Each of the dictionary elements is represented as a convex combination of a *small* (sparse) set of *representatives* that are real biological observations. The convex hull of these representatives is termed the *representative region* of the dictionary element. As a result, the vertices of the representative regions comprise a collection of MCMC-generated real-world samples. Figure 2a shows the organization of a dictionary as a collection of dictionary elements, representatives, and representative regions.

After initialization, we perform iterative optimization of the DL objective function using online samples, again generated via the MCMC method. More precisely, at each iteration, we compute the distance between the new sample and every current estimate of dictionary elements. Subsequently, we assign the sample to the representative region of the nearest dictionary element, which leads to an increase in the size of the set of representatives associated with the dictionary element. From this expanded set of representatives, we carefully select one representative for removal, maximizing the improvement in the quality of our dictionary element and the objective function. It is possible that the removed representative is the newly added data sample assigned to the representative region. In this case, the dictionary element remains unchanged. Otherwise, it is obtained as a convex combination of the updated set of representatives. After observing sufficiently many online samples, the algorithm converges to an accurate set of dictionary elements or the procedure terminates without convergence (in which case we declare a failure and restart the learning process). In our experiments, we never terminated with failure, but due to the lack of provable convergence guarantees for real-world datasets, such scenarios cannot be precluded. The update procedure is shown in Figure 2b.

We applied the method outlined above to RNAPII-enriched ChIA-Drop data from *Drosophila Melanogaster* S2 cells, using a dm3 reference genome [17], to learn dictionaries of chromatin interactions. Figure 3 provides an illustration of the ChIA-Drop pipeline.

We preprocessed the RNAPII ChIA-Drop data to remove fragments mapped to the repetitive regions in the genome and performed an MIA-Sig enrichment test with FDR 0.1 [32]. Only the hyperedges that passed this test were used in our subsequent analysis. To facilitate the analysis, we binned chromosomal genetic sequences into 500 bp regions and used the midpoint of each fragment for mapping. These bins of 500 consecutive bases form the nodes of the hypergraph for each chromosome, while the set of filtered multiway interactions form the hyperedges. The dataset hence includes $45,938$, $42,292$, $49,072$, and $55,795$ nodes and $36,140$, $28,387$, $53,006$, $45,530$ hyperedges for chromosome chr2L, chr2R, chr3L and chr3R respectively. The distribution of the hyperedge sizes is given in Supplement Table 1. To create networks from hypergraphs, we converted the multiway interactions into cliques. The clique-expanded input network has $113,606$, $85,316$, $161,590$, and $143,370$ edges respectively. Although the ChIA-Drop data comprises interactions from six chromosomes chr2L, chr2R, chr3L, chr3R, chr4 and chrX, since chr4 and chrX are relatively short regions and most of the functional genes are located on chr2L, chr2R, chr3L, and chr3R, we focus our experiments only on the latter.

In the experiments, we set the number of dictionary elements to $K = 25$. The number of dictionary elements K is selected to achieve the best trade-off between accuracy and complexity of the learned dictionary representations. Small values of K do not fully capture the diversity of multiway interactions present in the data, while very large values result in unnecessarily redundant representations. The latter can also obscure important interactions by capturing the inherent noise in ChIA-Drop data, and contribute to representation distortion [31]. After testing our method for multiple different values of K, we settled for $K = 25$. Clearly, other datasets may benefit from a different choice of the parameter K, which has to be fine-tuned. Also, as template

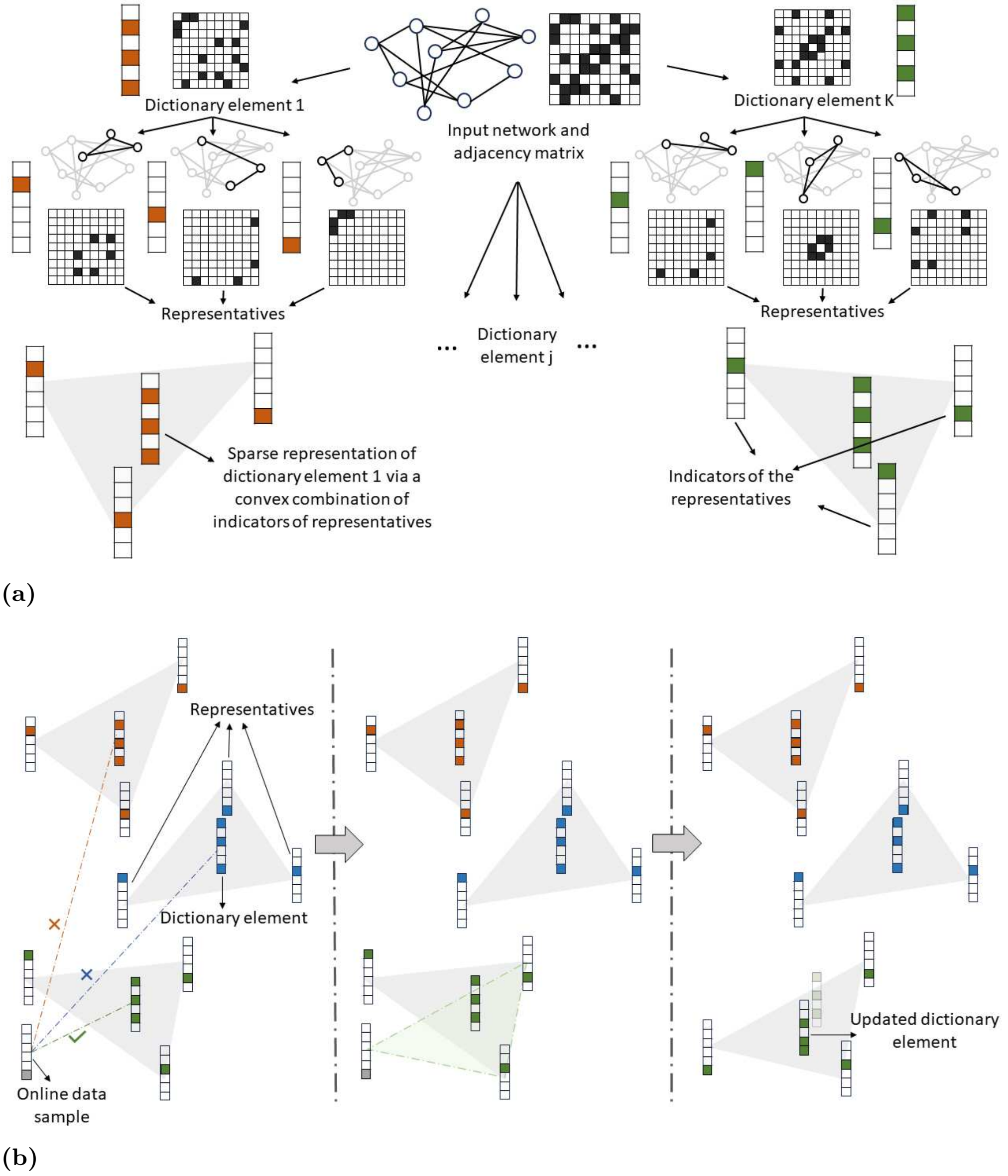
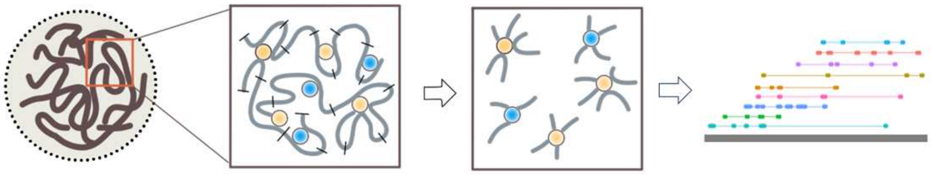

Dictionary element 1
Input network and adjacency matrix
Dictionary element K
Representatives
Representatives
...
Dictionary element j
...
Sparse representation of dictionary element 1 via a convex combination of indicators of representatives
Indicators of the representatives
(a)
Representatives
Dictionary element
Online data sample
Updated dictionary element
(b)

Fig 2. (a) Organization of a dictionary comprising K dictionary elements that are convex combinations of real representative subnetworks. Each dictionary element itself is a sparse *convex combination* of a set of representatives which are small subnetworks of the input real-world network. In the example, there are 6 options for the representatives, and inclusion of a representative into a dictionary element is indicated by a colored entry in a 6-dimensional indicator column-vector. Each of the 6 representatives corresponds to a subnetwork of the input network with a fixed number of nodes (3 for our example). The dictionary element is generated by a convex combination of the corresponding adjacency matrices of its corresponding representative subnetworks. For the example, the resulting dictionary elements are 9×9 matrices. (b) Illustration of the representative region update. When an online data sample is observed, the distance of the sample to each of the current dictionary elements is computed and the sample is assigned to the representative region of the nearest dictionary element. From this expanded set of representatives, one representative is carefully selected for removal to improve the objective. The new dictionary element is then obtained as an optimized convex combination of the updated set of representatives.



(a)

Fig 3. Generation of ChIA-Drop data. ChIA-Drop [17] adopts a droplet-based barcode-linked technique to reveal multiway chromatin interactions at a single molecule level. Chromatin samples are crosslinked and fragmented without a proximity ligation step. The samples are enriched for informative fragments through antibody pull-down.

subnetworks, we use *paths*, since paths are the simplest and most common network motifs, especially in chromatin interaction data (most contact measurements are proximal due to the linear chromosome order). Once again, by optimizing via trial-and-error, we select paths including 21 nodes (i.e., 21×500 bases). Both the choice of the subnetwork (motif) and its number of constituent nodes is data dependent.

MCMC sampling for initialization, as well as for subsequent online optimization steps, was performed before running the online optimization process to improve the efficiency of our implementation. We sampled $20,000$ subnetworks from each of the four chromosomes to ensure sufficient coverage of the input network. From this pool of subnetworks, we randomly selected 500 subnetworks to initialize our dictionaries, ensuring that each dictionary element had at least 10 representatives (which suffice to get quality initializations for the dictionary elements themselves). Each online step involved sampling an additional subnetwork and we iterated this procedure up to 1 million times, as needed for convergence (see Figure 1a).

At this point, it is crucial to observe that the dictionary elements learned by online cvxNDL effectively capture *long-range interactions* because each dictionary element may include distal genomic regions that are not adjacent in the genomic order. In other words, the diagonal entries of our dictionary elements *do not exclusively represent consecutive genomic regions* as in standard chromatin contact maps; instead, they may include *both* nonconsecutive (long-range) and consecutive (short-range, adjacent) interactions. This point is explained in detail in Figure 4. Another relevant remark is that without the convexity constraint, dictionary element entries could not have been meaningfully mapped back (associated) to genomic regions and viewed as *real physical interactions between genomic loci.*

The dictionary elements generated from the *Drosophila* ChIA-Drop data for chr2L, chr2R, chr3L, and chr3R using the online cvxNDL method are shown in Figure 5. Each subplot corresponds to one chromosome and has 25 dictionary elements ordered with respect to their *importance scores*, capturing the relevance and frequency of use of the dictionary element, and formally defined in the Methods Section. Each element is color-coded based on the genomic location of the genes covered by their representatives. Hence, dictionary elements represent combinations of experimentally observed interaction patterns, uniquely capturing the significance of the genomic locations involved in the corresponding interactions. We also report the density and median distance between all consecutive pairs of interacting loci (connected nodes) of all dictionary elements in Supplement Tables 2 and 3.

Note that our algorithm is the first method for online learning of convex (interpretable) network dictionaries. We can therefore only compare its *representation accuracy* to that of nonnegative matrix factorization (NMF), convex matrix

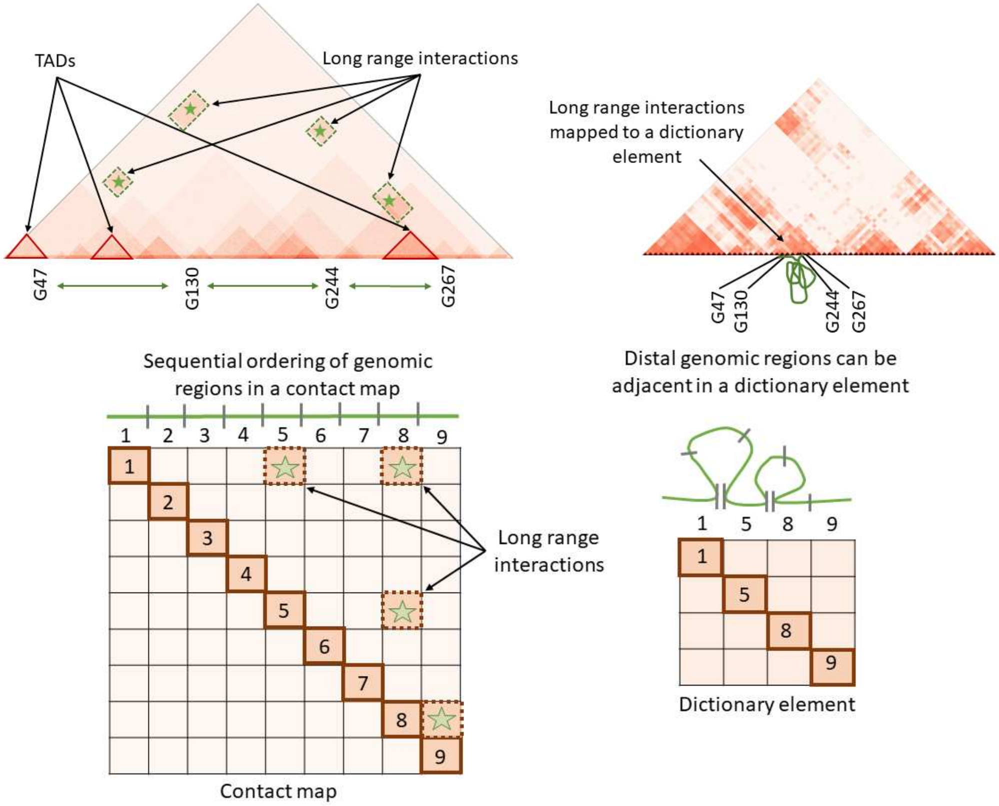

TADs
Long range interactions
G47
G130
G244
G267
Long range interactions mapped to a dictionary element
G47
G130
G244
G267
Sequential ordering of genomic regions in a contact map
1 2 3 4 5 6 7 8 9
Long range interactions
Contact map
Distal genomic regions can be adjacent in a dictionary element
1 5 8 9
Dictionary element

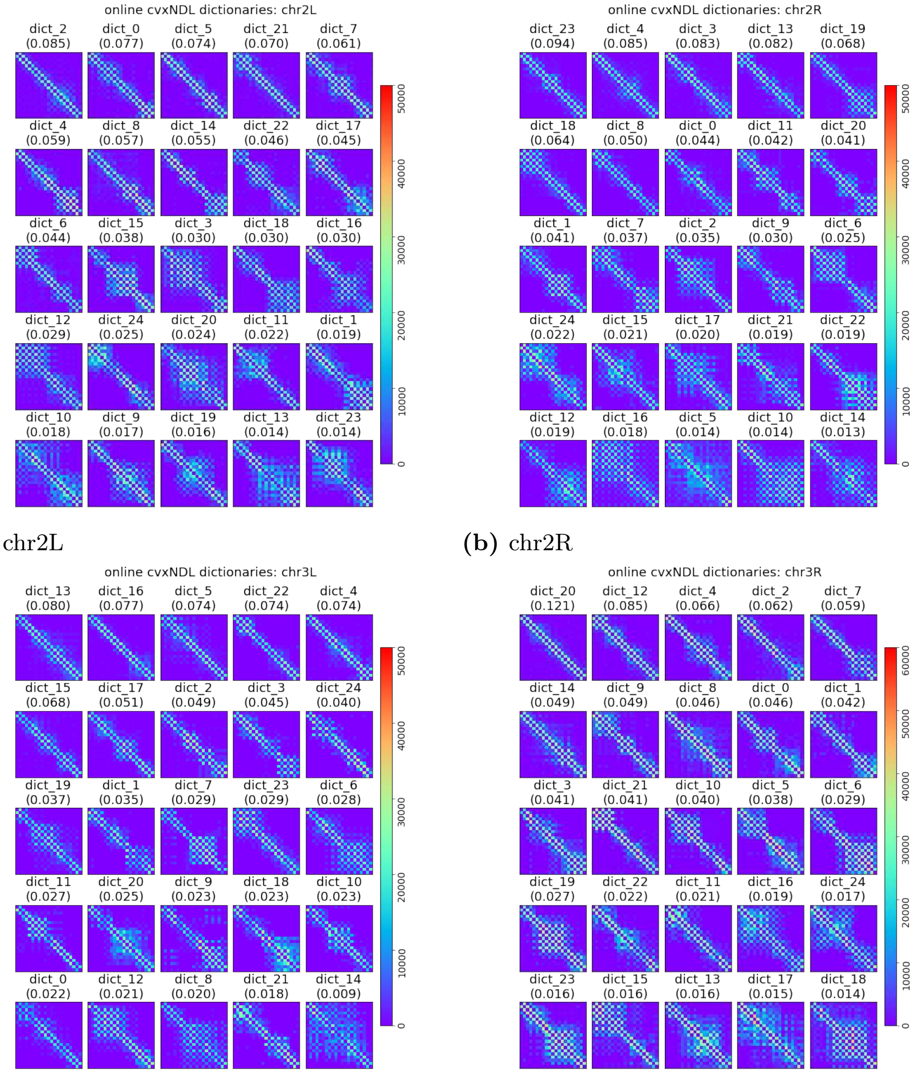
Fig 4. A dictionary element, represented as a matrix, consists of both proximal and distal interacting genomic regions. The elements on the diagonal are not necessarily indexed by adjacent (consecutive) genomic fragments, as explained by the example in the second row. There, off-diagonal long-range interactions in the 9×9 matrix are included in a 3×3 dictionary element whose diagonal elements are not in consecutive order.

factorization (CMF), and online network dictionary learning (online NDL). A visual comparison of the dictionaries formed through online cvxNDL and the aforementioned methods for chr2L is provided in Figure 6.

Classical NMF does not allow the mapping of results back to real interacting genomic regions. While the dictionary elements obtained via CMF are interpretable, they tend to mostly comprise widely spread genomic regions since they do not use the network information. The dictionary elements generated by online cvxNDL have smaller yet relevant spreads that are more likely to capture meaningful long-range interactions. In contrast to online cvxNDL, both NMF and CMF are not scalable to large datasets, rendering them unsuitable for handling current and future high-resolution datasets such as those generated by ChIA-Drop. Compared to online NDL, online cvxNDL also has a more balanced distribution of importance scores. For example, in Figure 6(b), dict_0 has score 0.459, while the scores in Figure 6(d) are all ≤ 0.085. Moreover, akin to standard NMF, NDL fails to provide interpretable results since the dictionary elements cannot be mapped back to real interacting genomic loci.

Results for other chromosomes are reported in the Supplement Section 4. Recall that both online cvxNDL and online NDL use a k-path as the template.

Reconstruction Accuracy. Once a dictionary is constructed, one can use the network reconstruction algorithm from [15] to recover a subnetwork or the whole



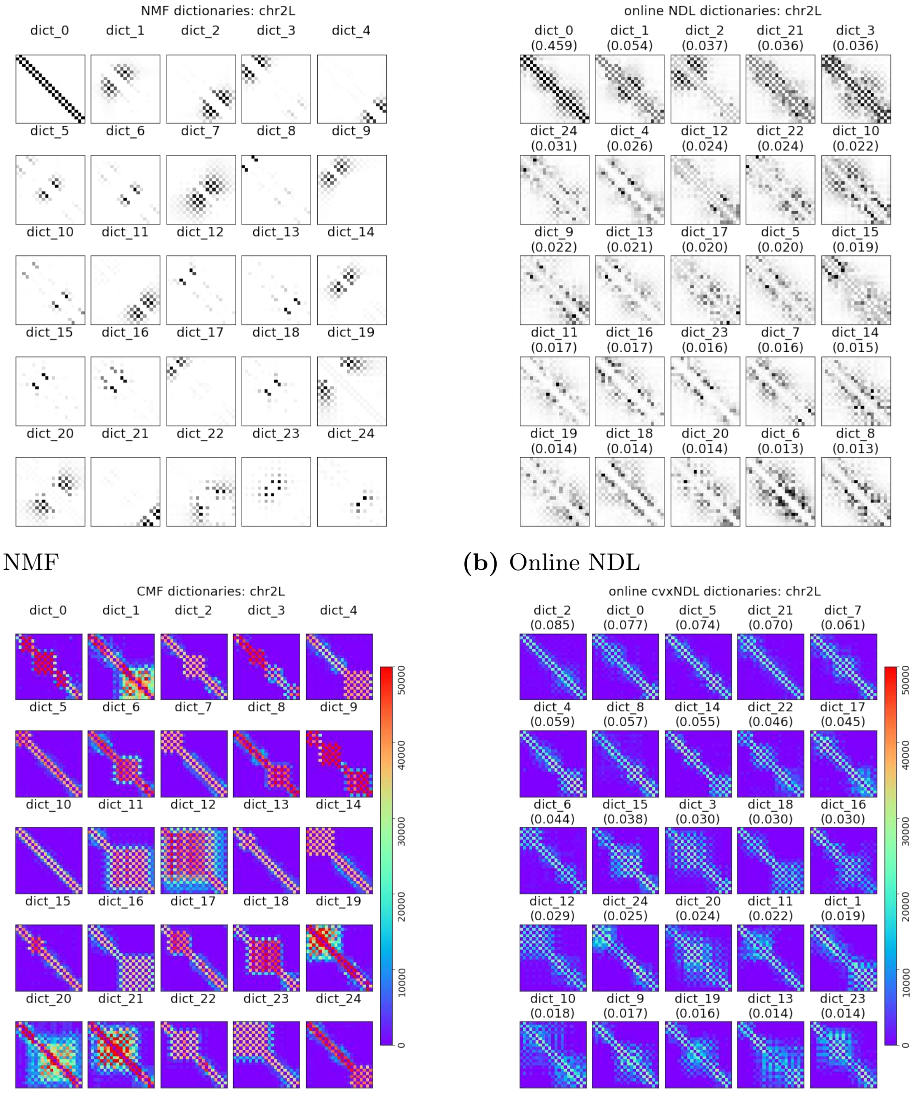
(a) chr2L

(b) chr2R

(c) chr3L

(d) chr3R

Fig 5. Dictionary elements for *Drosophila* chromosomes 2L, 2R, 3L and 3R obtained using online cvxNDL. Each subplot contains 25 dictionary elements for the corresponding chromosome and each block in the subplots corresponds to one dictionary element. The elements are ordered by their importance score. Note that the "diagonals" in the dictionary elements do not exclusively represent localized topologically associated domains (TADs) as in standard chromatin contact maps; instead, they can also capture long-range interactions. This is due to the fact that the indices of the dictionary element matrices represent genomic regions that may be far apart in the genome. In contrast, standard contact maps have indices that correspond to continuously ordered genomic regions, so that the diagonals truly represent TADs (see Figure 4). The color-code captures the actual locations of the genomic regions involved in the representatives and their dictionary elements. The most interesting dictionary elements are those that contain both dark blue and light blue/green and red spectrum colors (since they involve long-range interactions). This is especially the case for chr3L and chr3R.



(a) NMF

(b) Online NDL

(c) CMF

(d) Online cvxNDL

Fig 6. Dictionary elements for *Drosophila* chromosome chr2L generated by NMF (6a), online NDL (6b), CMF (6c) and online cvxNDL (6d). NMF and CMF are learned off-line, using a total of 20, 000 samples. Note that these algorithms do not scale and cannot work with larger number of samples such as those used in online cvxNDL. The color-coding is performed in the same manner as for the accompanying online cvxNDL results. Columns of the dictionary elements in the second row are color-coded based on the genome locations of the representatives. As biologically meaningful locations can be determined only via convex methods, the top row corresponding to NMF and online NDL results is black-and-white.

network by locally approximating subnetworks via dictionary elements. The accuracy of approximation in this case measures the "expressibility" of the dictionary with respect to the network. All methods, excluding randomly generated dictionaries used for illustrative purposes only, can accurately reconstruct the input network. For a quantitative assessment, the average precision-recall score for all methods is plotted in Table 1. As expected, random dictionaries have the lowest scores across all chromosomes, while all other methods are of comparable quality. This means that interpretable methods, such as our online cvxNDL, do not introduce representation distortions (CMF also learns interpretable dictionaries; however, it is substantially more expensive computationally when compared to our method but does not ensure that network topology is respected). A zoomed-in sample-based reconstruction result for chr2L is shown in Supplement Figure 6, while the reconstruction results for the entire contact maps of chr2L, chr2R, chr3L, and chr3R are available in Supplement Figures 7-10. Additionally, for synthetic data, Figure 7 shows the reconstructed adjacency matrices for various dictionary learning methods, further confirming the validity of findings for the chromatin data. More detailed results for synthetic data are available in Supplement Section 3.

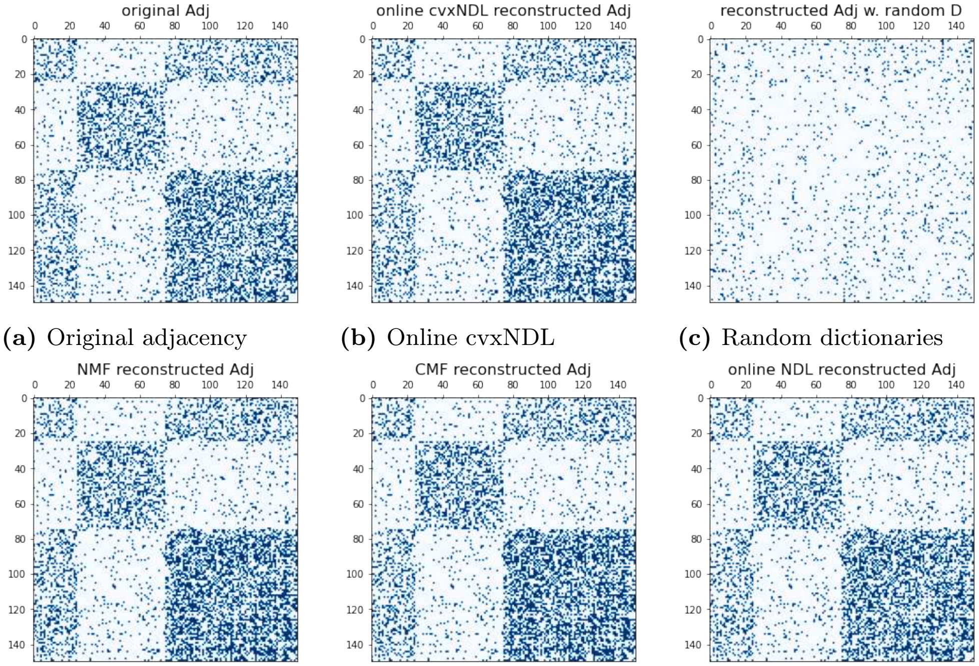
Table 1. Average Precision Recall for different DL methods, for all chromosomes as well as synthetic datasets. Methods that return interpretable dictionaries are indicated by the superscript i while methods that are scalable to large datasets are indicated by the superscript s. Online cvxNDL is both interpretable and scalable while maintaining performance on par with other noninterpretable and nonscalable methods.

	chr2L	chr2R	chr3L	chr3R	Synthetic
Online cvxNDL^{i,s}	0.9954	0.9986	0.9830	0.9876	0.9747
Online NDL^{s}	0.9955	0.9986	0.9834	0.9880	0.9728
NMF	0.9952	0.9985	0.9829	0.9873	0.9774
CMF^{i}	0.9951	0.9985	0.9824	0.9870	0.9731
Random Dict.	0.0007	0.2547	0.5276	0.0796	0.1922

Gene Ontology Enrichment Analysis. As each dictionary element is associated with a set of representatives that correspond to real observed subnetworks, their nodes can be mapped back to actual genomic loci. This allows one to create lists of genes covered by at least one node included in the representatives.

To gain insights into the functional annotations of the genes associated with the dictionary elements, we conducted a Gene Ontology (GO) enrichment analysis using the annotation category "Biological Process" from `http://geneontology.org`, with the reference list *Drosophila Melanogaster*. This analysis was performed for each dictionary element. Our candidate set for enriched GO terms was selected with a false discovery rate (FDR) threshold of < 0.05. Note that the background genes used for comparison are all genes from all chromosomes (the default option). We also utilized the hierarchical structure of GO terms [33], where terms are represented as nodes in a directed acyclic graph, and their relationships are described via arcs in the digraph (i.e., each "child" GO term is more specific than its "parent" term and where one child may have multiple parents).

We further refined our results by running additional processing steps. For each GO term, we identified all the paths between the term and the root node and then removed any intermediate parent GO term from the enriched GO terms set. By iteratively performing this filtering process for each dictionary element, we created a list of the most specific GO terms associated with each element. More details about the procedure are available in the Supplement Section 6.



(a) Original adjacency **(b)** Online cvxNDL **(c)** Random dictionaries

(d) NMF **(e)** CMF **(f)** Online NDL

Fig 7. Original adjacency matrix and reconstructed adjacency matrices based on different DL methods, including randomly selected dictionaries. The figure illustrates the fact that the additional convexity constraint does not compromise the quality of interaction representation/reconstruction in a visual manner. For more rigorous analytical accuracy comparisons Table 1.

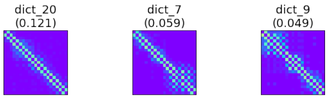
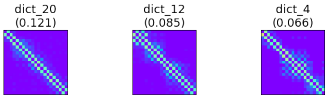
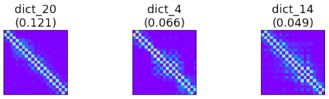
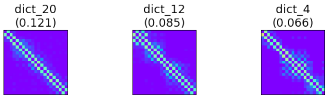
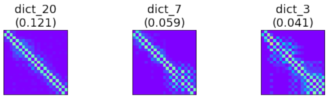
We report the most frequently enriched GO terms for each chromosome, along with the corresponding dictionary elements exhibiting enrichment for chr3R in Table 2. The results for other chromosomes are available in the Supplement Tables 4-6. Notably, the most frequent GO terms are related to regulatory functions, reflecting the significance of RNA Polymerase II. We also observe that dictionary elements for chr2L and chr2R are enriched in GO terms associated with reproduction and embryonic development. Similarly, chr3L and 3R are enriched in GO terms for blood circulation and responses to heat and cold.

We report the number of GO terms associated with each dictionary element, along with their importance scores in Supplement Tables 10-13. Dictionary elements with higher importance scores tend to exhibit a larger number of enriched GO terms while dictionary elements with 0 enriched GO terms generally have small importance scores.

RNA-Seq Coexpression Analysis. The ChIA-Drop dataset [17] used in our analysis was accompanied by a single noisy RNA-Seq replicate. To address this issue, we retrieved 20 collections of RNA-Seq data corresponding to untreated S2 cell lines of *Drosophila Melanogaster* from the Digital Expression Explorer (DEE2) repository. DEE2 provides uniformly processed RNA-Seq data sourced from the publicly available NCBI Sequence Read Archive (SRA) [23]. The list of sample IDs is available in Supplement Table 14.

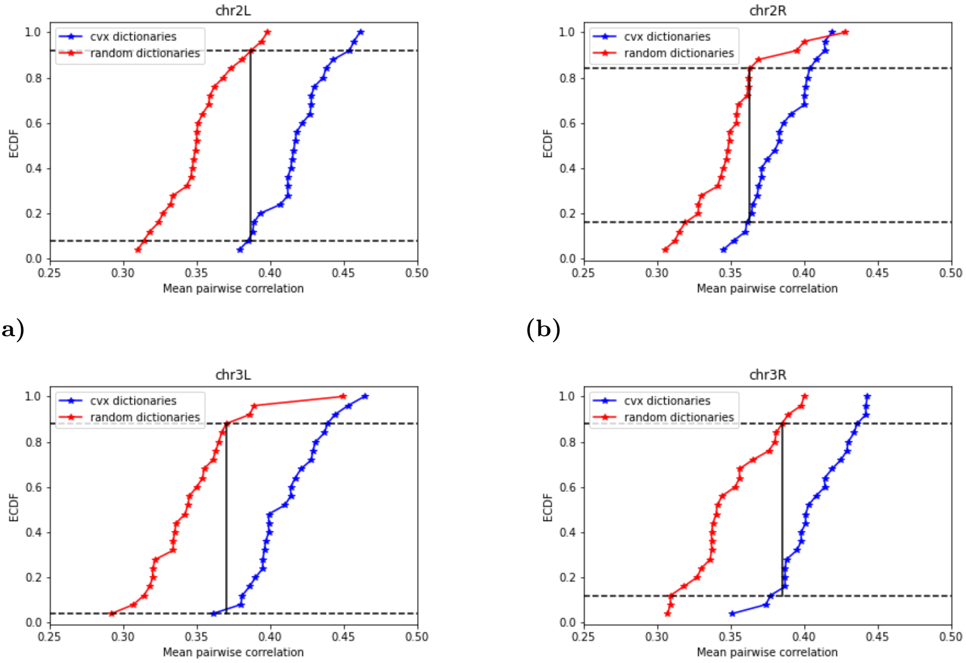
To ensure consistent normalization across all samples, we used the trimmed mean of M values (TMM) method [34], available through the edgeR package [35]. This is of crucial importance when jointly analyzing samples from multiple sources. We selected the most relevant genes by filtering the list of covered genes and retaining only those

Table 2. The 5 most enriched GO terms for genes covered by dictionary elements from chr3R. Column '#' indicates the number of dictionary elements that show enrichment for the given GO term. Also reported are up to 3 dictionary elements with the largest importance score in the dictionary, along with the "density" ρ of interactions in the dictionary element (defined in the Methods section) and median distance d_{med} of all adjacent pairs of nodes in its representatives.

Most frequent GO term	#	Top 3 dictionaries
(GO:0001819) Positive regulation of cytokine production	7	 ρ=0.126,0.146,0.157 d_{med}=12791,12830,11930
(GO:0008015) Blood circulation	7	 ρ=0.126,0.142,0.138 d_{med}=12791,13455,13674
(GO:0045948) Positive regulation of translational initiation	5	 ρ=0.126,0.138,0.162 d_{med}=12791,13674,12572
(GO:0042177) Negative regulation of protein catabolic process	5	 ρ=0.126,0.142,0.138 d_{med}=12791,13455,13674
(GO:0043065) Positive regulation of apoptotic process	4	 ρ=0.126,0.146,0.179 d_{med}=12791,12830,11748

with more than 95% overlap with the gene promoter regions, as defined in the *Ensembl* genome browser. Subsequently, for each dictionary element, we collected all genes covered by it and then calculated the pairwise Pearson correlation coefficient of expressions of pairs of genes in the set. To visualize the underlying coexpression clusters within the genes, we performed hierarchical clustering, the results of which are shown in Supplement Section 7 and Figure 9. The latter corresponds to the R1-R4 and T1-T4 genomic regions of chr2L to be discussed in what follows.

Additionally, we conducted control experiments by constructing dictionary elements through random sampling of genes from the list of all genes on each of the chromosomes. For these randomly constructed dictionaries, we carried out a coexpression analysis as described above. We observed that the mean of coexpressions of all pairs of genes in a randomly constructed dictionary element is significantly lower compared to the mean of the online cvxNDL dictionary elements. Specifically, for dictionary elements generated using online cvxNDL, the mean coexpression values for all pairs of genes covered by the 25 dictionary elements, and for each of the four chromosomes, 2L, 2R, 3L, and 3R, were found to be $0.419, 0.383, 0.411$, and 0.407, respectively. The corresponding values for randomly constructed dictionaries were found to be $0.333, 0.329, 0.323$, and 0.337, respectively. To determine if these differences are statistically significant, we employed the two-sample Kolmogorov-Smirnov test [36], comparing the empirical cumulative distribution functions (ECDFs) of pairwise coexpression values of the learned and randomly constructed dictionaries. The null hypothesis used was "the two sets of dictionary elements are drawn from the same underlying distribution.” The null hypotheses for all four chromosomes were rejected, with p-values equal to 3.6×10^{-9}, 8.5×10^{-6}, 3.6×10^{-9}, and 2.5×10^{-7} for chr2L, chr2R, chr3L, and chr3R, respectively. This indicates that the learned dictionary elements indeed capture meaningful biological patterns of chromatin interactions.



(a)

(b)

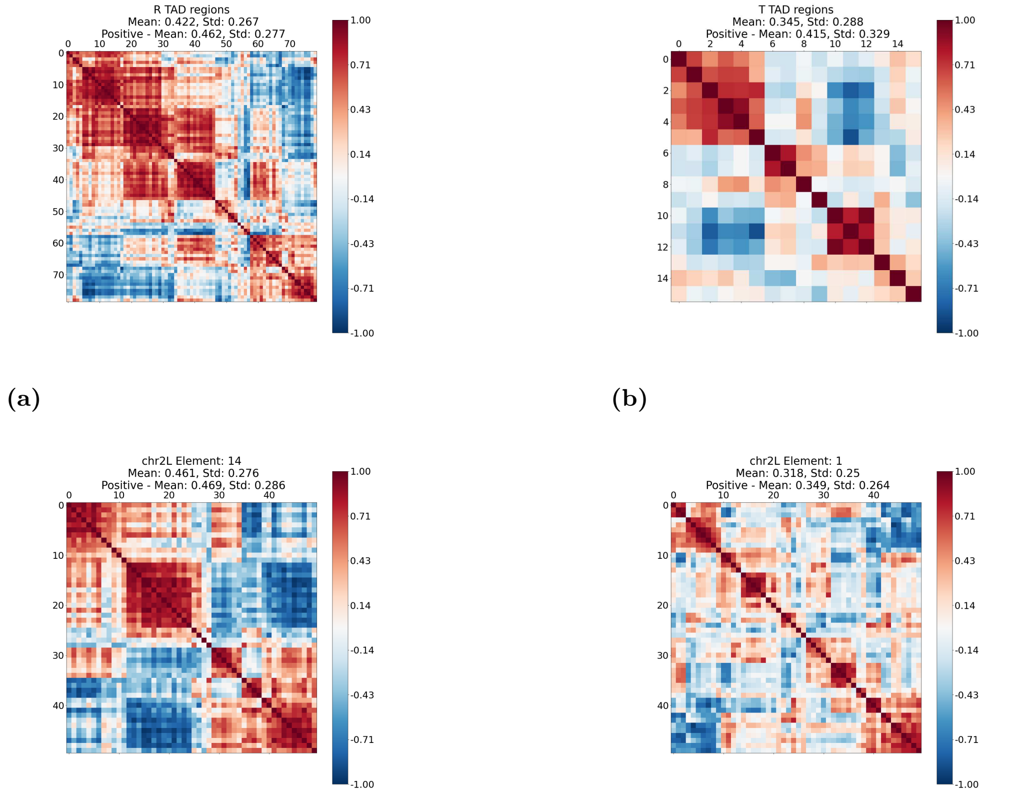
(c)

(d)

Fig 8. Empirical cumulative distribution functions (ECDF) of mean pairwise coexpressions of genes covered by random and online cvxNDL dictionary elements ((a) for chr2L, (b) for chr2R, (c) for chr3L and (d) for chr3R). The results are based on the two-sample Kolmogorov-Smirnov test, and the null hypothesis described in the main text.

To further evaluate our results, we also examined the well-documented R1-R4 and T1-T4 TAD interactions on chr2L, reported in [17]. The results of the coexpression analysis for these genomic regions are reported in Figure 9. The mean pairwise correlation between genes belonging to the R1-R4 genomic regions equals 0.422, which is comparable to the mean value 0.419 of the results obtained via online cvxNDL. We also calculated the intersection of the set of genes within the R1-R4 genomic regions and the set of genes covered by online cvxNDL dictionary elements identified for chr2L. We observed that the top 5 online cvxNDL dictionary elements cover 38 out of 85 genes in the R1-R4 genomic regions. This is to be contrasted with the results for random dictionary elements, which cover only 7 genes. Table 3 describes these and related findings in more detail.

Finally, we mapped genes covered by our dictionary elements onto nodes of the STRING protein-protein interaction network [24]. These mappings allow us to determine the confidence of pairwise gene interactions. These, and related results based on FlyMine [37] data, a large gene expression repository for *Drosophila Melanogaster*, are available in Supplement Section 8.



(a) **(b)**

(c) **(d)**

Fig 9. Pairwise coexpression of genes covered by (a) the R1-R4 genomic regions, (b) the T1-T4 genomic regions, (c) an online cvxNDL dictionary element, and (d) a randomly constructed dictionary element. We calculated the mean and standard deviation of absolute pairwise coexpression values, and the mean and standard deviation of coexpression values specifically for all positively correlated gene pairs. The mean coexpression values within TADs and dictionary elements are similar to each other and generally higher than those of randomly constructed dictionary elements. Note that the plot (b) is of coarser resolution due to the small number of genes covered when compared to the cases (a), (c), (d).

	Online cvxNDL			Random		
	Dictionary element id	Intersection	Cumulative	Dictionary element id	Intersection	Cumulative
1	1	15	15	20	3	3
2	11	12	24	0	1	4
3	12	12	30	1	1	5
4	7	11	35	21	1	6
5	21	10	38	17	1	7

Table 3. Intersection between the set of genes within the R1-R4 genomic regions and the sets of genes covered by online cvxNDL dictionary elements for chr2L. We determined the sizes of the intersections of the set of genes covered by each dictionary element and the genes in the R1-R4 genomic region and arranged them in decreasing order. The top 5 dictionary elements in this order cumulatively contain 38 out of the 85 genes within the R1-R4 genomic regions. This is in sharp contrast with randomly generated dictionary elements, where the top 5 elements with maximum intersection cover only 7 genes.

Methods

Notation. Sets of consecutive integers are denoted by $[l] = \{1, \ldots, l\}$. The symbol $\mathbb{N}$ is reserved for the natural numbers. Capital letters are reserved for matrices (bold font) and random variables (RVs) (regular font). Vectors are denoted by lower-case underlined letters. For a matrix of dimension $d \times n$ over the reals, $\mathbf{A} \in \mathbb{R}^{d\times n}$, $\mathbf{A}[i,:]$ is used to denote the i^{th} row and $\mathbf{A}[:,i]$ the i^{th} column of $\mathbf{A}$. The entry in row i, column j is denoted by $\mathbf{A}[i,j]$. Similarly, $\underline{x}[l]$ is used to denote the l^{th} coordinate of a deterministic vector $\underline{x} \in \mathbb{R}^d$. Furthermore, we use the standard notation for the ℓ_1 and Frobenius norm of matrices, $\|\mathbf{A}\|_1 = \sum_{i,j} |\mathbf{A}[i,j]|$ and $\|\mathbf{A}\|_F^2 = \sum_{i,j} \mathbf{A}[i,j]^2$, respectively.

A network $\mathcal{G} = ([n], \mathbf{A})$ is an ordered pair of sets, the node set $[n]$, and the set of edges represented by their adjacency matrix $\mathbf{A}$. Our underlying assumption is that the network is connected, which means that every node can be reached from every other node. Also, $\mathbf{A}[i,j] = \mathbf{A}[j,i] \in \{0,1\}$, indicating the presence or absence of an undirected edge between nodes i, j. In addition, $\text{Col}(\mathbf{A})$ stands for the set of columns of $\mathbf{A}$, while $\text{cvx}(\mathbf{A})$ stands for the convex hull of $\text{Col}(\mathbf{A})$.

Online DL. We first formulate the online DL problem. Assume that N input data samples are generated by a random process and organized in matrices $(\mathbf{X}_t)_{t\in\mathbb{N}} \in \mathbb{R}^{d\times N}$ indexed by time t. For $N = 1$, $\mathbf{X}_t$ reduces to a column vector that encodes a d-dimensional signal. Given an online, sequentially observed data stream $(\mathbf{X}_t)_{t\in\mathbb{N}}$, the goal is to find a sequence of dictionary matrices $(\mathbf{D}_t)_{t\in\mathbb{N}}, \mathbf{D}_t \in \mathbb{R}^{d\times K}$, and codes $(\mathbf{\Lambda}_t)_{t\in\mathbb{N}}, \mathbf{\Lambda}_t \in \mathbb{R}^{K\times N}$, such that when $t \to \infty$ almost surely we have

$$\|\mathbf{X}_t - \mathbf{D}_t\mathbf{\Lambda}_t\|_F^2 \to \min_{\mathbf{D},\mathbf{\Lambda}} \mathbb{E}_{\mathbf{X}} \|\mathbf{X} - \mathbf{D}\mathbf{\Lambda}\|_F^2. \tag{1}$$

The expected loss in Equation 1 can be minimized by iteratively updating $\mathbf{\Lambda}_t$ and $\mathbf{D}_t$ every time a new data sample $\mathbf{X}_t$ is observed. The approximation error of $\mathbf{D}$ for a single data sample $\mathbf{X}$ is chosen as

$$l(\mathbf{X}, \mathbf{D}) = \min_{\mathbf{\Lambda}\in\mathbb{R}^{K\times N}} \|\mathbf{X} - \mathbf{D}\mathbf{\Lambda}\|_F^2 + \lambda \|\mathbf{\Lambda}\|_1 . \tag{2}$$

The second term represents a sparsity-enforcing regularizer. Furthermore, the empirical f_t and surrogate loss $\hat{f}_t$ for $\mathbf{D}$ are defined as

$$f_t(\mathbf{D}) = (1 - w_t)f_{t-1}(\mathbf{D}) + w_t l(\mathbf{X}_t, \mathbf{D}), t \geq 1, \tag{3}$$

$$\hat{f}_t(\mathbf{D}) = (1 - w_t)\hat{f}_{t-1}(\mathbf{D}) + w_t(\|\mathbf{X}_t - \mathbf{D}\mathbf{\Lambda}\|_F^2 + \lambda \|\mathbf{\Lambda}\|_1), \tag{4}$$

where the weight w_t determines the sensitivity of the algorithm to the newly observed data. The online DL algorithm first updates the code matrix $\mathbf{\Lambda}_t$ by solving Equation (2) with $l(\mathbf{X}_t, \mathbf{D}_{t-1})$, then updates the dictionary matrix $\mathbf{D}_t$ by minimizing (4) via

$$\mathbf{D}_t = \underset{\mathbf{D} \in \mathbb{R}^{d \times r}}{\arg\min}(\operatorname{Tr}(\mathbf{D}\mathbf{A}_t\mathbf{D}^T) - 2\operatorname{Tr}(\mathbf{D}\mathbf{B}_t)), \tag{5}$$

where $\mathbf{A}_t = (1 - w_t)\mathbf{A}_{t-1} + w_t\mathbf{\Lambda}_t\mathbf{\Lambda}_t^T$ and $\mathbf{B}_t = (1 - w_t)\mathbf{B}_{t-1} + w_t\mathbf{\Lambda}_t\mathbf{X}_t^T$ are the aggregated history of the input data and their codes, respectively. For simplicity, we set $w_t = \frac{1}{t}$.

To add convexity constraints, we introduce for each dictionary element a *representative set (region)* $\hat{\mathbf{X}}_t^{(i)} \in \mathbb{R}^{d \times N_i}, i \in [K]$, where N_i is the size of the representative set for dictionary element $\mathbf{D}_t[:, i]$, and $N = \sum_{i=1}^K N_i$. The representative set for a dictionary element is a small subcollection of real data samples observed up to time t that best explain the dictionary element they are assigned to. The set of representatives is updated after observing a sample, the inclusion of which provides a better estimate of the dictionary element compared to the previous set. Since the representative set is bounded in size, if a new sample is included, an already existing sample has to be removed (see Figure 2b). Formally, the optimization objective is of the form

$$\min_{\mathbf{D} \in \operatorname{cvx}(\hat{\mathbf{X}}), \hat{\mathbf{X}}} \hat{f}_t(\mathbf{D}) = \min_{\mathbf{D} \in \operatorname{cvx}(\hat{\mathbf{X}}), \hat{\mathbf{X}}} \left(1 - \frac{1}{t}\right) \hat{f}_{t-1}(\mathbf{D}) + \frac{1}{t}\left(\|\mathbf{X}_t - \mathbf{D}\mathbf{\Lambda}_t\|_F^2 + \lambda \|\mathbf{\Lambda}_t\|_1\right). \tag{6}$$

MCMC sampling of subnetworks (sample generation). For NDL, it is natural to let the columns of $\mathbf{X}_t$ be vectorized adjacency matrices of N subnetworks. Hence one needs to efficiently sample meaningful subnetworks from a (large) network. In image DL problems, samples can be generated directly from the image using adjacent rows and columns. However, such a sampling technique cannot be applied to arbitrary network data. Selecting nodes along with their one-hop neighbors at random may produce subnetworks of vastly different sizes and the results do not capture meaningful long-range interactions. It is also difficult to trim such subnetworks to uniform sizes. Furthermore, sampling a fixed number of nodes uniformly at random from sparse networks produces disconnected subnetworks with high probability and is not an acceptable approach either.

To address these problems, we consider "subnetwork sampling" introduced in [14, 15] where we fix a template network $F = ([k], \mathbf{A}_F)$ of k nodes and seek subnetworks induced by k nodes in the input network $\mathcal{G}$, with the constraint that the subnetwork *contains* (but does not necessarily equals) the template F topology. Given an input network $\mathcal{G} = ([n], \mathbf{A})$ and a template network $F = ([k], \mathbf{A}_F)$, we define a set of homomorphisms as a vector of the form

$$\texttt{Hom}(F, \mathcal{G}) = \left\{ \underline{x} : [k] \to [n] \,\middle|\, \prod_{1 \leq i,j \leq k} \mathbf{A}[\underline{x}[i], \underline{x}[j]]^{\mathbf{A}_F[i,j]} = 1 \right\}, \tag{7}$$

where we by default assume that $0^0 = 1$. For each homomorphism $\underline{x} \in \texttt{Hom}(F, \mathcal{G})$, denote its induced adjacency matrix by $\mathbf{A}_{\underline{x}}$, where $\mathbf{A}_{\underline{x}}[a, b] = \mathbf{A}[\underline{x}[a], \underline{x}[b]]$,

$1 \leq a, b \leq k$. The adjacency matrix $\mathbf{A}_{\underline{x}}$ represents one sample from the input network $\mathcal{G}$. An example homomorphism is shown in Figure 1d, where the input network $\mathcal{G}$ contains $n = 9$ nodes and the template network F is a star network that contains $k = 4$ nodes. One proper homomorphism in this case is $\underline{x}[a] = 9$, $\underline{x}[b] = 6$, $\underline{x}[c] = 4$, $\underline{x}[d] = 7$, which gives rise to an adjacency matrix $\mathbf{A}_{\underline{x}}$ as depicted. A homomorphism can be sampled using the rejection sampling algorithm presented in the Supplement Section 2, Algorithm 1. Our choice of template network, as already mentioned, is a *k-path*, i.e., a path joining k nodes. Paths are a simple and natural choice for networks with long average path lengths, such as chromatin interaction networks. It is also the same choice of template used in standard NDL. As a final remark, we note that a k-path homomorphism leads to a sample of dimension $d = k^2$, as we will flatten its $k \times k$ adjacency matrix into a single vector.

Although rejection sampling can be used repeatedly to generate several homomorphisms, it is highly inefficient. To efficiently generate a sequence of sample adjacency matrices $\mathbf{A}_{\underline{x}_t}$ from $\mathcal{G}$, the MCMC sampling algorithm is used instead, while rejection sampling is only used to initialize the MCMC algorithm.

Next, for a homomorphism $\underline{x}_t$, let $\mathcal{N}[\underline{x}_t[1]]$ ($\mathcal{N}$ for short) denote the set of neighbors of $\underline{x}_t[1]$. We first choose a node $v \in \mathcal{N}$ from the neighborhood of $\underline{x}_t[1]$ uniformly at random, i.e. with probability $P(v) = \frac{1}{|\mathcal{N}|}$. We also calculate the probability of acceptance β for the selected node v. For a $k-$path template used in our approach, the value of β is given by

$$\beta = \min \left\{ \frac{\sum_{c \in [n]} A^{k-1}[v, c]}{\sum_{c \in [n]} A^{k-1}[\underline{x}_t[1], c]} \frac{\sum_{c \in [n]} A[\underline{x}_t[1], c]}{\sum_{c \in [n]} A[v, c]}, 1 \right\}, \tag{8}$$

following the guidelines from [14, 15].

Next, we draw a value $u \in [0, 1]$ uniformly at random. If $u < \beta$, we accept $\underline{x}_{(t+1)}[1] = v$, otherwise we reject v and reset $\underline{x}_{(t+1)}[1] = \underline{x}_t[1]$. We then perform a directed random walk from $\underline{x}_{t+1}[1]$ of length equal to $k - 1$ to obtain $\underline{x}_{(t+1)}[2], \ldots, \underline{x}_{(t+1)}[k]$. An illustration of the sampling procedure is shown in Figure 1e, while the detailed algorithm is presented in the Supplement Section 2, Algorithm 2.

Online convex NDL (online cvxNDL). We start by initializing the dictionary $\mathbf{D}_0$ and representative sets $\{\hat{\mathbf{X}}_0^{(i)}\}, i \in [K]$, for each dictionary element. The algorithm for initialization is presented in the Supplement Section 2 Algorithm 3. After initialization, we perform iterative optimization to generate $\mathbf{D}_t$ and $\{\hat{\mathbf{X}}_t^{(i)}\}, i \in [K]$, to reduce the loss at round t. At each iteration, we use MCMC sampling to obtain a k-node random subnetwork as sample $\mathbf{X}_t$, and then update the codes Λ_t based on the dictionary $\mathbf{D}_{t-1}$ by solving the optimization problem in Equation (2). Then we assign the current sample to a representative set of the closest dictionary element, say $\mathbf{D}_{t-1}[:, j]$, and jointly update its representative set $\hat{\mathbf{X}}_t^{(j)}$ and all dictionaries $\mathbf{D}_t$ as shown in Figure 2b. The iterative update algorithm for online cvxNDL is presented in the Supplement Section 2 Algorithm 4.

The output of the algorithm is a dictionary matrix $\mathbf{D}_T \in \mathbb{R}^{k^2 \times K}$, where each column is a flattened vector of a dictionary element of size $k \times k$, and the representative sets $\{\hat{\mathbf{X}}_T^{(i)}\}, i \in [K]$, for each dictionary element. Each representative set $\hat{\mathbf{X}}_T^{(i)} \in \mathbb{R}^{k^2 \times N_i}$ contains N_i history-sampled subnetworks from the input network as its columns which are called the representatives of the dictionary element. The convex hull of all representatives of a dictionary element forms the representative region of the dictionary element. We can easily convert both the dictionary elements and representatives back to $k \times k$ adjacency matrices. Due to the added convexity constraint, each dictionary element $\mathbf{D}_T[:, j]$ at the final step T has the *interpretable* form:

$$\mathbf{D}_T[:,j] = \sum_{i\in[N_j]} w_{j,i}\hat{\mathbf{X}}_T^{(j)}[:,i], \quad \text{s.t.} \sum_{i\in[N_j]} w_{j,i} = 1, w_{j,i} \geq 0, i \in [N_j], j \in [K]. \tag{9}$$

The weight $w_{j,i}, i \in [N_j]$, is the *convex coefficient* of the i^{th} representative of dictionary element $\mathbf{D}_T[:,j]$. Dictionary elements learned from the data stream can be used to reconstruct the input network by multiplying it with the dictionary element weights from Equation (2). The j^{th} index of the weight vector corresponds to the contribution of dictionary element $\mathbf{D}_{T-1}[:,j]$ to the reconstruction. Similarly to what was done in [15], we can also define the *importance score* for each dictionary element as

$$\gamma(i) = \frac{\mathbf{A}_t[i,i]^2}{\sum_{j\in[K]}\mathbf{A}_t[j,j]^2}. \tag{10}$$

We use the importance scores, as described in the previous sections, to determine the most frequently used interactions in the dictionary construction, as well as the most typical and important long-range interactions.

To conclude, we point out that the *density* ρ of interactions in a dictionary element is defined as

$$\rho = \frac{1}{k^2}\sum_{i,j=1}^{k}\mathbf{D}_T[i,j].$$

Funding and Acknowledgement

The work was supported by the National Science Foundation grants #1956384 and #2206296 and grant CZI DAF2022 − 249217. The authors gratefully acknowledge many useful discussions with Dr. Yijun Ruan.

Supporting information

Supplemental material, including figures and tables, is available in the Supplement file. The online cvxNDL code and test datasets are available at:
`https://github.com/rana95vishal/chromatin_DL/`

References

1. Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image processing. 2006;15(12):3736–3745.

2. Mairal J, Elad M, Sapiro G. Sparse representation for color image restoration. IEEE Transactions on image processing. 2007;17(1):53–69.

3. Cichocki A, Lee H, Kim YD, Choi S. Non-negative matrix factorization with α-divergence. Pattern Recognition Letters. 2008;29(9):1433–1440.

4. Ye M, Qian Y, Zhou J. Multitask sparse nonnegative matrix factorization for joint spectral–spatial hyperspectral imagery denoising. IEEE Transactions on Geoscience and Remote Sensing. 2014;53(5):2621–2639.

5. Lu H, Sang X, Zhao Q, Lu J. Community detection algorithm based on nonnegative matrix factorization and pairwise constraints. Physica A: Statistical Mechanics and its Applications. 2020;545:123491.

6. Zhu X, Ching T, Pan X, Weissman SM, Garmire L. Detecting heterogeneity in single-cell RNA-Seq data by non-negative matrix factorization. PeerJ. 2017;5:e2888.

7. Shao C, Höfer T. Robust classification of single-cell transcriptome data by nonnegative matrix factorization. Bioinformatics. 2017;33(2):235–242.

8. Zhang S, Chasman D, Knaack S, Roy S. In silico prediction of high-resolution Hi-C interaction matrices. Nature communications. 2019;10(1):1–18.

9. Paatero P, Tapper U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics. 1994;5(2):111–126.

10. Paatero P. Least squares formulation of robust non-negative factor analysis. Chemometrics and intelligent laboratory systems. 1997;37(1):23–35.

11. Ding CH, Li T, Jordan MI. Convex and semi-nonnegative matrix factorizations. IEEE transactions on pattern analysis and machine intelligence. 2010;32(1):45–55.

12. Mairal J, Bach F, Ponce J, Sapiro G. Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research. 2010;11(Jan):19–60.

13. Peng J, Milenkovic O, Agarwal A. Online convex matrix factorization with representative regions. In: Advances in Neural Information Processing Systems; 2019. p. 13242–13252.

14. Lyu H, Memoli F, Sivakoff D. Sampling random graph homomorphisms and applications to network data analysis. Journal of machine learning research. 2023;24(9):1–79.

15. Lyu H, Needell D, Balzano L. Online matrix factorization for Markovian data and applications to Network Dictionary Learning. Journal of Machine Learning Research. 2020;21(251):1–49.

16. Lyu H, Kureh YH, Vendrow J, Porter MA. Learning low-rank latent mesoscale structures in networks. To appear in Nature Communications. arXiv preprint arXiv:210206984. 2021;.

17. Zheng M, Tian SZ, Capurso D, Kim M, Maurya R, Lee B, et al. Multiplex chromatin interactions with single-molecule precision. Nature. 2019;566(7745):558–562.

18. Li G, Ruan X, Auerbach RK, Sandhu KS, Zheng M, Wang P, et al. Extensive promoter-centered chromatin interactions provide a topological basis for transcription regulation. Cell. 2012;148(1-2):84–98.

19. Tang Z, Luo OJ, Li X, Zheng M, Zhu JJ, Szalaj P, et al. CTCF-mediated human 3D genome architecture reveals chromatin topology for transcription. Cell. 2015;163(7):1611–1627.

20. Lieberman-Aiden E, Van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. science. 2009;326(5950):289–293.

21. Li G, Fullwood MJ, Xu H, Mulawadi FH, Velkov S, Vega V, et al. ChIA-PET tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. Genome biology. 2010;11(2):R22.

22. Fullwood MJ, Liu MH, Pan YF, Liu J, Xu H, Mohamed YB, et al. An oestrogen-receptor-α-bound human chromatin interactome. Nature. 2009;462(7269):58–64.

23. Ziemann M, Kaspi A, El-Osta A. Digital expression explorer 2: a repository of uniformly processed RNA sequencing data. Gigascience. 2019;8(4):giz022.

24. Szklarczyk D, Gable AL, Lyon D, Junge A, Wyder S, Huerta-Cepas J, et al. STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic acids research. 2019;47(D1):D607–D613.

25. Wang S, Zhang Q, He Y, Cui Z, Guo Z, Han K, et al. DLoopCaller: A deep learning approach for predicting genome-wide chromatin loops by integrating accessible chromatin landscapes. PLoS Computational Biology. 2022;18(10):e1010572.

26. Xie WJ, Qi Y, Zhang B. Characterizing chromatin folding coordinate and landscape with deep learning. PLoS computational biology. 2020;16(9):e1008262.

27. Zhang P, Wu Y, Zhou H, Zhou B, Zhang H, Wu H. CLNN-loop: a deep learning model to predict CTCF-mediated chromatin loops in the different cell lines and CTCF-binding sites (CBS) pair types. Bioinformatics. 2022;38(19):4497–4504.

28. Tian SZ, Li G, Ning D, Jing K, Xu Y, Yang Y, et al. MCIBox: a toolkit for single-molecule multi-way chromatin interaction visualization and micro-domains identification. Briefings in Bioinformatics. 2022;23(6):bbac380.

29. Agarwal S, Lim J, Zelnik-Manor L, Perona P, Kriegman D, Belongie S. Beyond pairwise clustering. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2. IEEE; 2005. p. 838–845.

30. Zhou D, Huang J, Schölkopf B. Learning with hypergraphs: Clustering, classification, and embedding. Advances in neural information processing systems. 2006;19.

31. Li P, Milenkovic O. Inhomogeneous hypergraph clustering with applications. Advances in neural information processing systems. 2017;30.

32. Kim M, Zheng M, Tian SZ, Lee B, Chuang JH, Ruan Y. MIA-Sig: multiplex chromatin interaction analysis by signal processing and statistical algorithms. Genome biology. 2019;20(1):1–13.

33. Musen MA. The protégé project: a look back and a look forward. AI matters. 2015;1(4):4–12.

34. Robinson MD, Oshlack A. A scaling normalization method for differential expression analysis of RNA-seq data. Genome biology. 2010;11(3):1–9.

35. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. bioinformatics. 2010;26(1):139–140.

36. Massey Jr FJ. The Kolmogorov-Smirnov test for goodness of fit. Journal of the American statistical Association. 1951;46(253):68–78.

37. Lyne R, Smith R, Rutherford K, Wakeling M, Varley A, Guillier F, et al. FlyMine: an integrated database for Drosophila and Anopheles genomics. Genome biology. 2007;8(7):1–16.