Node2Vec on Hi-C: Detecting Chromosome Translocations with Network Analysis

Author: Keivan Amini

Abstract

In recent years, advancements in genomics have led to the development of high-resolution techniques such as Hi-C, which enable the exploration of spatial proximity and interactions between genomic loci [1]. This project aims to elucidate the potential genetic transmutations between distinct chromosomes through a comprehensive analysis pipeline. The study leverages Hi-C matrices as adjacency matrices to construct a network representation of chromosomal interactions, followed by the application of the Node2Vec algorithm [2] to learn embeddings capturing complex chromatin interactions. Furthermore, Principal Component Analysis (PCA) [3] is employed to reduce dimensionality and visualize the embeddings in a lower-dimensional space. To facilitate and automate this process, the Node2VecHiC library has been developed using Python [4].

1 Introduction

Within this project, we focused on a particular type of genomic data obtained through the Hi-C technique, which aims to capture the three-dimensional spatial organisation of chromatin within a cell's nucleus.

Building upon the foundational insights offered by Hi-C data, our study extends the analysis to explore genetic transmutations between selected chromosome pairs. The Node2Vec algorithm, a versatile graph embedding technique, is employed to capture the latent features of chromosomal interactions within the Hi-C matrix-derived network. Recently, Node2Vec has demonstrated its efficacy in extracting meaningful embeddings from complex networks [5], enabling the representation of intricate relationships for tasks such as community detection. The proposed approach follows recent trends in applying network-based methodologies to biological data, for uncovering hidden patterns and functional associations.

Moreover, we harness the power of Principal Component Analysis (PCA) to further dissect and visualize the acquired embeddings. PCA is a widely used dimensionality reduction technique that has proven invaluable in revealing underlying structural patterns in complex datasets. By projecting the learned embeddings onto a lower-dimensional space, we aim to provide a clear and interpretable visualization of the potential genetic transmutations between chromosomes.

In the subsequents sections of this project, we detail our methodology for constructing the Hi-C network, exploiting the Node2Vec algorithm, and conducting PCA analysis. We focus on investigating genetic interactions in a tumore cell, between chromosome 6 and chromosome X, as well as between chromosome 10 and chromosome 20, as illustrative examples. Furthermore, we discuss the implications of our findings, highlighting the potential for extending this approach to unravel transmutations across the entire genome.

2 Hi-C data

In this section we discuss the origin of the Hi-C data, which are usually obtained by biologists performing the experiment shown in Figure 2.

The DNA strands contained within the cell nucleus are all folded close together and with formaldehyde, the DNA strands that are close in space are fixed. Artificially then these strands are cut once fixed and then bound: in the Figure 2 it is possible to see two different strands, one blue and one red, which are two segments of DNA that, if loose and extended according to the strand, could also be very far apart along the strand, but in space are close together as they are folded.

Artificially, this new DNA strand is created. The subsequent processes are the fragmentation of DNA and then sequencing. In the sequencing, a reference genome that represents the standard describing the average state of a healthy person is considered: on this reference genome, a mapping[1] is carried out.

The artificial fragments that are produced in the experiment are marked, and both the blue and the red fragments are mapped onto the reference genome.

With this procedure, it is possible to construct a contact matrix M, in which each entry corresponds to the number of times a pair of fragments has been seen close together in the space. The obtained contact matrix is squared ($M_{n \times n}$) and symmetrical ($M = M^T$). Usually, these experiments are conducted on a population of 1 million or 5 million cells.

In this project, we worked on two Hi-C matrices contained in `.csv` format, respectively associated with a healthy cell and a tumour cell. These matrices refer to a subset of the entire genome: data associated with chromosomes 1, 6, X, 10 and 20 are reported. In addition, an excel file was provided containing metadata, i.e. information about the different chromosomes corresponding to the different rows and columns of the source matrix.

From previous experimental data, a rearrangement of genomic material in the tumour cell was observed for chromosome pairs 6-X and 10-20. The 6-X translocation, as it emerges in the Hi-C matrix, can be described not as an exchange of material between chromosomes (i.e., some nodes on chromosome 10 were given to chromosome 20 and vice versa), but rather as a sharing of nodes (i.e., some nodes on chromosome 6 belong to both chromosome 6 and chromosome X and vice versa). Instead, there is a symmetrical exchange of genetic material between chromosome 10 and chromosome 20, whereby chromosome 10 is cut off and reattached to chromosome 20, as can be seen in Figure 1.

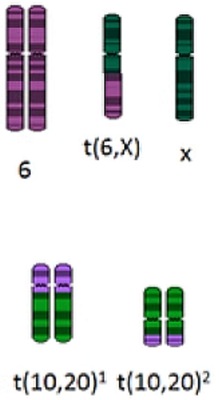

Figure 1: Translocations experimentally observed in the analyzed tumor cell.

The central idea of the project is to apply the Node2Vec algorithm to these matrices and then a PCA to study the extent to which these genetic translocations can be identified. Another approach is to work on pairs of chromosomes: i.e. to consider only the submatrix relating to all possible pairs, and to apply the algorithms on individual pairs, to see if this can help detect abnormalities.

One can then consider extending the work to all the chromosomes that make up the genome.

3 Methods

3.1 Node2Vec

Node2Vec is a semi-supervised algorithm for scalable learning in networks [2]. The algorithm has been inspired by prior works on natural language processing [6], for which the aim is to optimize a graph-based objective function using stochastic gradient descent method (SGD).

The Node2vec algorithm returns feature representations that maximize the likelihood of network neighborhoods of nodes, in a $d-$dimensional feature space. To explore diverse neighborhoods of a given node, a family of biased random walks can be exploited. This modern algorithm is flexible, and offers tunable parameters useful to control over the search space.

[1] Mapping means consider the DNA strand in order to understand the relation with the reference sequence.

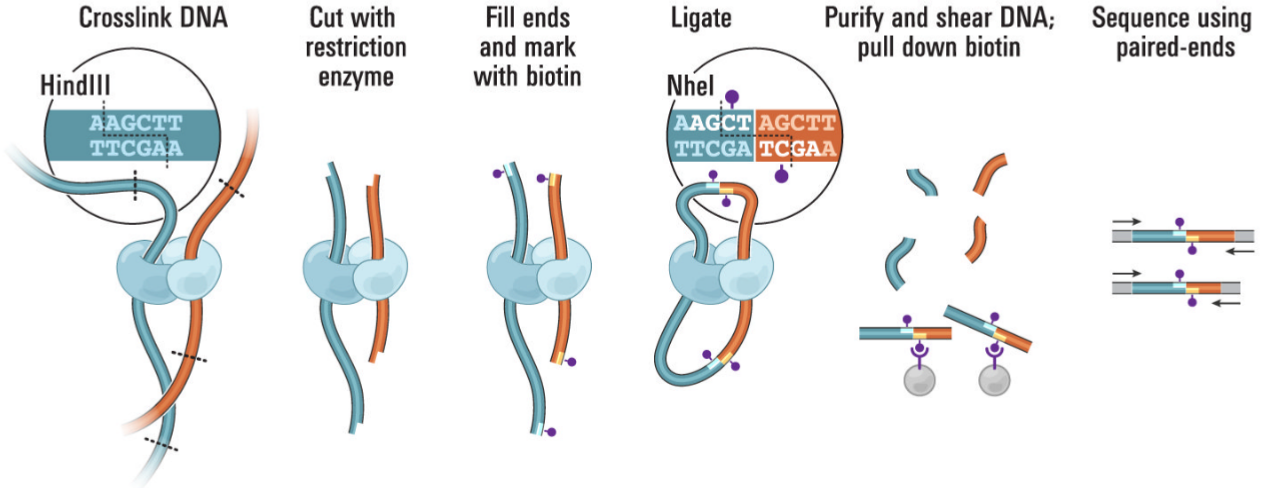
Crosslink DNA
Cut with restriction enzyme
Fill ends and mark with biotin
Ligate
Purify and shear DNA; pull down biotin
Sequence using paired-ends
HindIII
AAGCTT
TTCGAA
NheI
AAGCT AGCTT
TTCGA TCGAA

Figure 2: Cross-linking of chromatin segments that are in close spatial proximity, followed by the fragmentation of DNA, ligation of cross-linked fragments, and subsequent sequencing. Credits: [1].

Formally speaking, the feature learning problem is addressed as a maximum likelihood optimization problem. Let us define $G = (V, E)$ as a network with V vertices and E edges. Let $f : V \to \mathbb{R}^d$ be the mapping function from nodes to feature representations it is aimed to learn for a prediction task. Moreover, for each source node $u \in V$, a network neighborhood $N_S(u) \subset V$ it is defined. Here, S represents the specific neighborhood sampling strategy: clearly different strategies define different neighborhood.

The idea is to optimize the following objective function:

$$\max_f \sum_{u \in V} \log Pr(N_S(u)|f(u)) \qquad (1)$$

which maximizes the log-probability of observing a network neighborhood for a node u, conditioned on its feature representation. Here, two assumptions must be considered: the conditional independence of the presented probability, i.e. $Pr(N_S(u)|f(u) = \prod_{n_i} Pr(n_i|f(u))$, and the symmetry in feature space, for which it is possible to model the conditional likelihood $Pr(n_i|f(u))$ as a softmax unit parametrized by a dot product of their features. Given that, the Equation 1 becomes:

$$\max_f \sum_{u \in V} \left[-\log Z_u + \sum_{n_i} f(n_i) \cdot f(u) \right] \qquad (2)$$

where $Z_u = \sum_v \exp(f(u) \cdot f(v))$ represents the per-node partition function. As previously said, it is possible to optimize this problem by implementing stochastic gradient ascent over the model parameters defining the features f.

In order to design a flexible neighborhood sampling strategy, a biased random walk procedure can be implemented. Let us consider a random walk: the source node is u and let c_i be the ith node in the walk. In the proposed algorithm, the nodes c_i are generated by:

$$P(c_i = x|c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

where the term π_{vx} is the transition probability between nodes v and x, while the divisor Z represents the normalizing constant. This formula computes the probability of choosing the node x in the walk, given to the fact that the previous node was v. If (v, x) is an edge of the graph G, then logically the probability of choosing x is the normalized probability to transit from v to x. Otherwise, if v and x are not connected, the probability of choosing x as the next node of the random walk is worth zero.

Moreover, the authors of the algorithm decided to bias the random walk, specifying a way to define the unnormalized transition probability π_{vx}, with the introduction of two parameters p and q, which physical meaning can be observed in Figure 3.

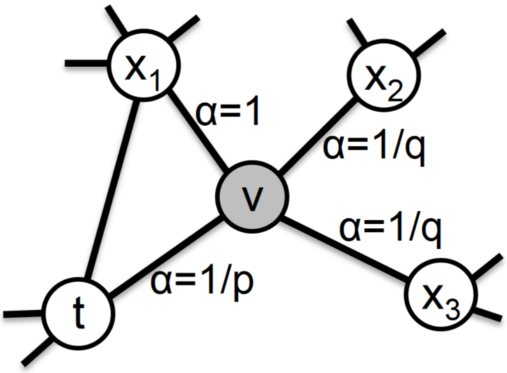

Figure 3: Graphical representation of the random walk. The walk came from node t, and now is evaluating the next step to take in node v, which can be x_1, x_2 or x_3. The parameter p and q guide the walk. Credits: [2]

In this biased scenario, we have the following definition:

$$\pi_{vx} = \alpha_{pq}(t,x) \cdot w_{vx} \tag{4}$$

where w_{vx} is the the static edge weights, that equal 1 for unweighted graphs. Moreover, the function $\alpha_{pq}(t,x)$ is given by:

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \tag{5}$$

where d_{tx} represents the shortest path distance between nodes t and x. Parameters p and q control how fast the walk explores the neighborhood of starting node u. Specifically, p controls the likelihood of immediately revisiting a node in the walk. If we set it to a large value, it is less likely to sample an already visited node in the following two steps; otherwise if p is small, it would lead the walk to backtrack a step, making the walk local. On the other side, if the parameter $q > 1$, the random walk is biased towards nodes close to node t, and if $q < 1$, the walk is inclined to visit nodes which are away from node t.

Summing up the Node2Vec algorithm, to obtain feature representations the procedure is the following:

1. Preprocessing transition probabilites
2. Random walk simulations
3. Optimization using SGD

Within this project, the algorithm was used by exploiting a Python Node2Vec library previously found on GitHub [7]. Here, the function `node2vec` can take in input different parameters such as the embedding dimensions d, the number of nodes in each walk w_l, the number of walks per node n_w, the return hyper parameter p, inout parameter q and the key for the weight attribute. The embedded results will be stored in a `.csv` data frame, which will be examined by exploiting a Principal Component Analysis, which makes it possible to reduce the dimensionality of the statistical problem addressed.

4 Principal Component Analysis

PCA is a commonly used method to examine extensive datasets that have many dimensions or features for each piece of information. Its purpose is to enhance our understanding of data while maintaining as much valuable information as possible. This technique also empowers us to depict complex data, which exists in multiple dimensions, in a way that is easier to grasp [8].

At its core, PCA seeks to transform high-dimensional data into a new coordinate system—a set of orthogonal axes called principal components. These principal components are designed to capture the maximum variance present in the original data, with the first component explaining the most variance, followed by the second, and so forth. By reorienting the data in this manner, PCA enables the identification of patterns, relationships, and trends that might have been concealed within the original dataset's multidimensional space.

Formally speaking, let us consider a set of p features $X_1, .., X_p$. The first principal component of this set of features is the normalized linear combination of the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p \tag{6}$$

that has the largest variance. Clearly, by normalized it is understood that the relation

$\sum_j \phi_{j1}^2 = 1$ is true. Usually, the elements $\phi_{11}, ..., \phi_{p1}$ are named *loadings* of the first principal component, and the principal component loading vector is simply given by $\phi_1 = (\phi_{11}\ \phi_{21}\ ...\ \phi_{p1})^T$. Then, we look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip} \qquad (7)$$

that has the largest sample variance, again with the normalization constraint for which $\sum_j \phi_{j1}^2 = 1$. Finally, the first principal component loading vector solves the following optimization problem:

$$\max_{\phi_{11},...,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1}x_{ij} \right)^2 \right\} \qquad (8)$$

subject to the constraint $\sum_{j=1} \phi_{j1}^2 = 1$. The objective we are maximizing is just the sample variance of the n values of z_{i1}.

Eventually, this problem can be addressed using the matrix formalism:

$$\text{Var}(Z_1) = \text{Var}(X\phi_1) = \phi_1^T \text{Var}(X)\phi_1 \qquad (9)$$

In this framework, the optimization problem becomes:

$$\max_{\phi_1} \left\{ \phi_1{}^T \Sigma_{\phi_1} \right\} \qquad (10)$$

where $\Sigma \equiv \text{Var}(X)$ for the sake of simplicity, and the normalization constraint now takes the form of $\phi_1^T\phi_1 = 1$. Using the theory of Lagrange Multipliers (the complete demonstration can be found in [9]), solving the optimization problem one ends up with the following equation:

$$\Sigma\phi_1 = \lambda_1\phi_1 \qquad (11)$$

for which we can see the relationship between the eigenvalues and the eigenvectors of the covariance matrix Σ: λ_1 is an eigenvalue of Σ and ϕ_1 the corresponding eigenvector. Multiplying both sides by ϕ_1^t and using equation 9, we end up with:

$$\phi_1^T\Sigma\phi_1 = \lambda \qquad (12)$$

and that means λ_1 exactly coincides with $\text{Var}(Z_1)$, i.e. the quantity we want to maximize.

In conclusion, in order to derive the linear combination having the largest variance, we simply need to consider the largest eigenvalue of Σ and ϕ_1 will be the corresponding eigenvector. After the first principal component Z_1 of the features has been determined, we can find the second principal component Z_2, which will be the linear combination of the original features that has maximal variance out of all linear combinations that are orthogonal to the first principal component Z_1. This process will lead to solve the problem $\Sigma\phi_2 = \lambda_2\phi_2$, where λ_2 is an eigenvalue of Σ and ϕ_2 is the corresponding eigenvector. Here, we will choose the second largest eigenvalue of Σ and the corresponding eigenvalue.

The described process can be continued for all principal components: generally the k-th principal component of X is $Z_k = X\phi_k$, and $\text{Var}(Z_k) = \lambda_k$, i.e. the k-th largest eigenvalue of Σ.

5 Results

In PCA, the goal is to reduce the dimensionality of the statistical problem at hand. By assigning colors to chromosomes, it is possible to observe that in the healthy cell line, there are no reshufflings of genetic material, whereas conversely, this is observed for some chromosomes with overlaps of genetic material.

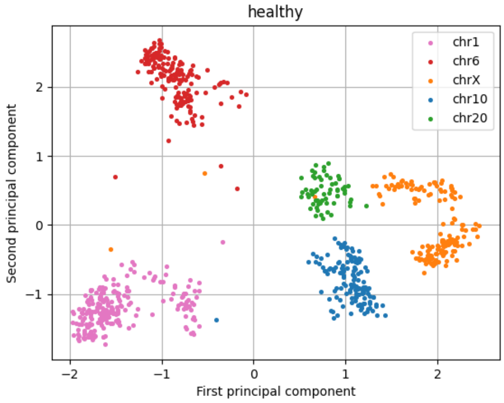

Figure 4: Principal component analysis on 10-dimensional embeddings learned with the Node2Vec algorithm. The production of distinct clusters without overlaps suggests qualitatively that there is no genetic material translocation between chromosomes.

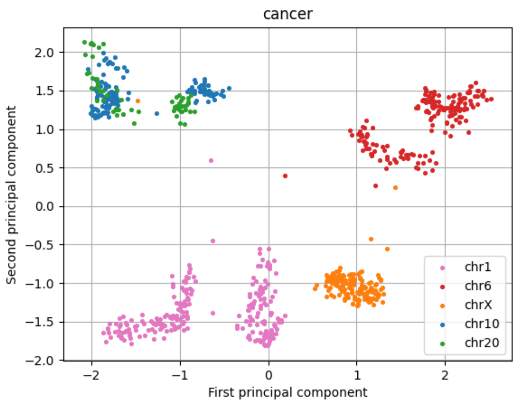

Figure 6: In the case of cancer cells, it is plausible to hypothesize that there was a genetic material exchange between chromosome 10 and chromosome 20, which has also been experimentally confirmed.

We started with an exploratory analysis adressed to the whole graphs structure both for the healthy and the cancer cell. Thus we performed the Node2Vec algorithm with the following parameters: $d = 10$, $n_w = 10$, $w_l = 10$, $p = 1$, $q = 0.5$.

It is important to emphasize that, from an experimental perspective, the absence of chromosomal translocations in the healthy cell has been confirmed. Concurrently, the presence of chromosomal translocations in the tumor cell has been experimentally verified, specifically concerning the pairs 10-20 and X-6.

As can be observed in Figure 6, only one of the two chromosomal translocations has been detected using this technique: no translocation between chromosome 6 and chromosome X is observed. To observe this specific genetic translocation using this approach, attempts were made to manipulate the various parameters of the involved algorithms.

5.1 Increasing embeddings dimension

Let us exam the influence of the parameter d on the resulting reduced embeddings.

Theoretically, when a smaller value is set for d, the embeddings will have fewer dimensions. Consequently, when PCA is applied, the dimensionality is further reduced. This can result in a more pronounced loss of information but can simplify data and facilitate more concise visualization. Conversely, when a larger d value is selected, it increases the number of dimensions in the embeddings. In such cases,

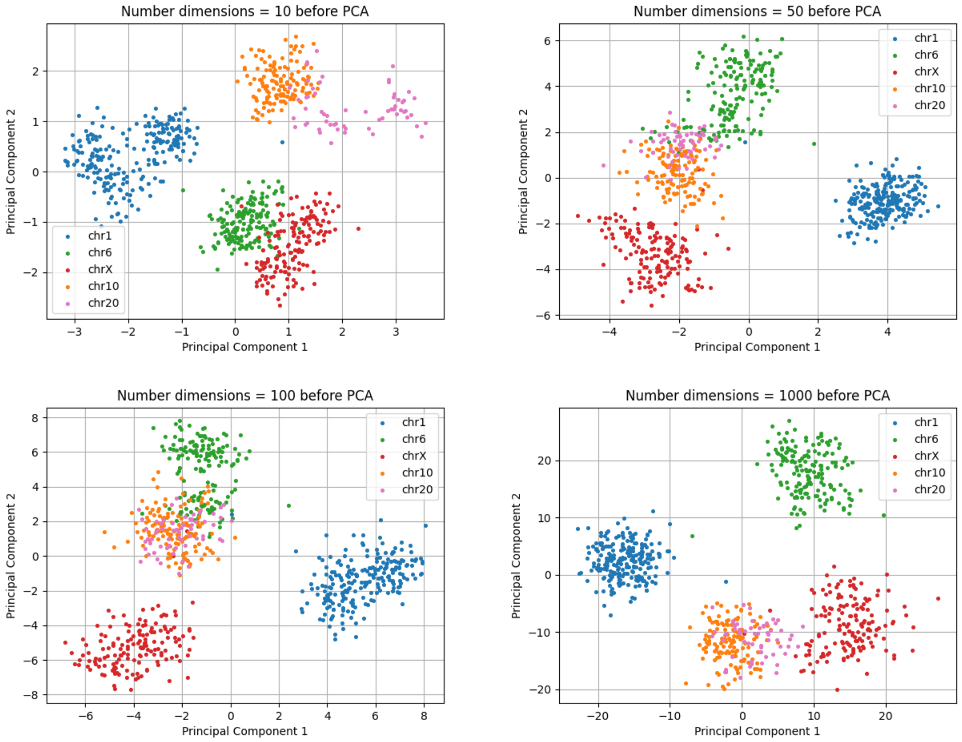

Figure 5: PCA on Node2Vec embeddings with different values of the parameter d.

when PCA is applied, dimensionality is still reduced, but a greater portion of the original information is retained.

In Figure 5, no major differences can be seen in the distribution of scatter points. The 10-20 translocation is visible in all the results presented; somewhat less visible in the case where $d = 10$. As far as the 6-X translocation is concerned, this is never observed; only in the case $d = 10$ is a slight overlap between the clusters associated with the two chromosomes observed.

5.2 Isolating chromosomes

Another technique for trying to observe 6-X translocation was to divide the cancer graph G_c into many graphs, for each possible chromosome pair. We first considered a selected chromosome and we constructed a list of graphs $[G_1, .., G_n]$ containing, in turn, the index nodes associated with the selected chromosomes and another chromosome. Specifically, if e.g the selected chromosome is chromosome 6, we would obtain a list of four graphs:

$$[G(\text{chr}_6, \text{chr}_1), ..., G(\text{chr}_6, \text{chr}_{20})]$$

By adopting this approach, we aimed to thoroughly examine the interactions and spatial arrangements of the selected chromosome with respect to every other chromosome present in the cell. Then, we repeated the process by changing the selected chromosome, in order to see the perfromance also with respect to chromosome we do not expect translocations. The results can be seen in Figure 11, contained in Appendix.

5.3 Varying number walks and walk length

The number of walks parameter n_w controls the number of random walks performed on the graph for each node during the embedding generation process. Specifically, it determines how many times the algorithm starts a random walk from each node. Each random walk explores the neighborhood of a node by traversing the graph along edges.

A reduction reduction in n_w signifies a decrease in the number of random walks initiated from each node. Consequently, this limits the exploration of local neighborhoods within the graph during embedding generation. As a re-

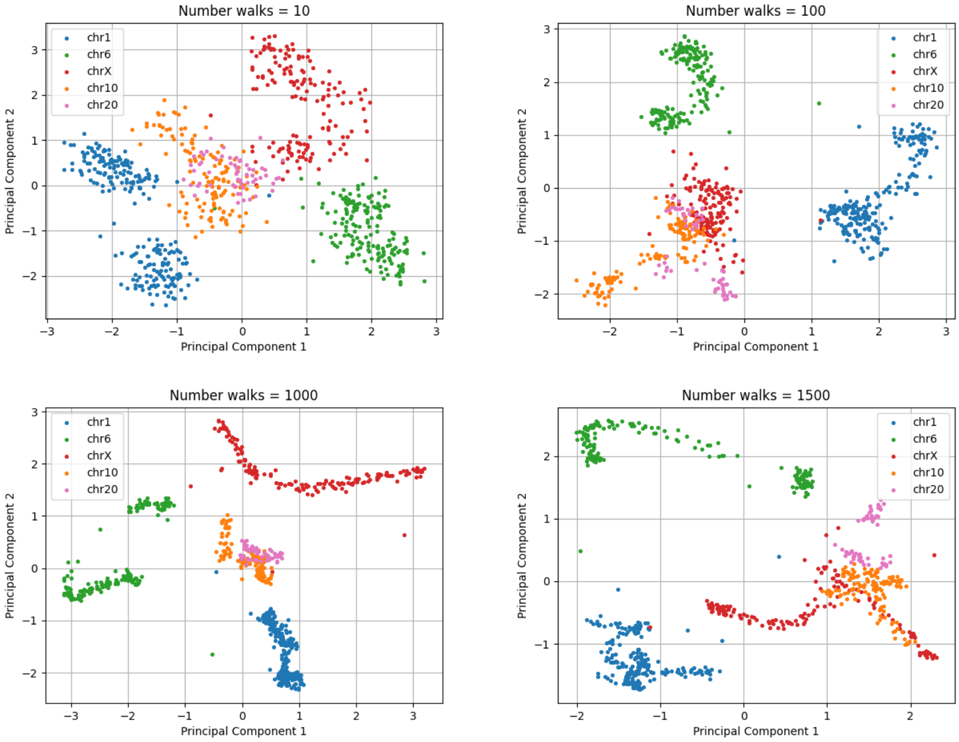

Figure 7: PCA on Node2Vec embeddings with different values of the parameter n_w

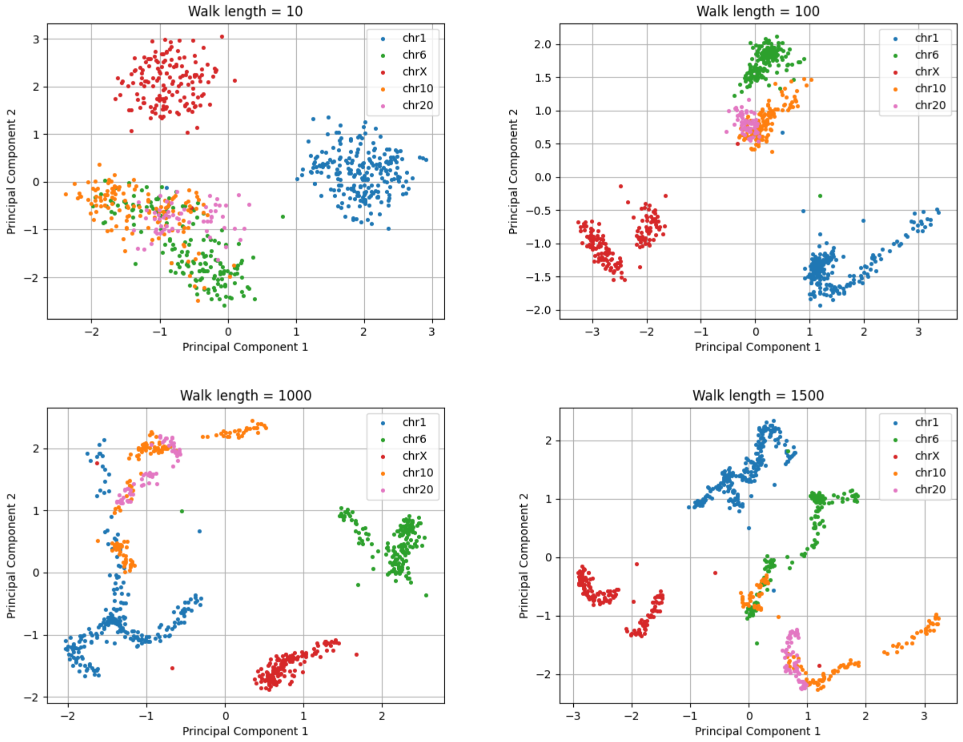

Figure 8: PCA on Node2Vec embeddings with different values of the parameter w_l.

sult, the resulting embeddings may fail to capture the finer details of the local graph structure. In PCA plots, this curtailed exploration can translate into reduced separation or differentiation between nodes, yielding less detailed visualizations.

Conversely, increasing n_w involves conducting a greater number of random walks starting from each node. This approach leads to a more exhaustive examination of the local neighborhoods within the graph. Consequently, embeddings generated with this setting are more likely to capture intricate details of the graph's structure. In PCA plots (see Figure 7), the heightened exploration can result in more pronounced distinctions between nodes, enhancing the level of detail and informativeness in the visualization.

On the other side, reducing w_l limits the exploration of local neighborhoods during embedding generation, resulting in embeddings that may miss finer details in the graph's structure, as seen in PCA plots (see Figure 8). Instead, increasing w_l leads to more extensive exploration, capturing intricate facets of the graph's structure. This heightened exploration results in more pronounced distinctions between nodes, enhancing detail and informativeness in the visualization.

5.4 Varying p and q parameters

When we adjust parameter p, we are essentially fine-tuning the scope of our exploration within the graph. A higher value of p encourages more focused local exploration, akin to the depth-first search (DFS) strategy. This results in embeddings that tend to favor capturing the nuances of the nearby graph structures. Conversely, a lower p opens the door to more exploratory walks, resembling the breadth-first search (BFS) strategy, potentially allowing us to capture a broader spectrum of characteristics embedded in the graph. This adjustment of parameter p can introduce variations in our PCA visualizations, where higher p values may give rise to well-defined clusters and intricate local patterns, similar to the outcomes of DFS, while lower p values might lead to visualizations with scattered and less clearly defined clusters, akin to the exploratory nature of BFS. Turning our attention to parameter q, a higher setting for q prompts the algorithm to venture into diverse and less connected regions of the

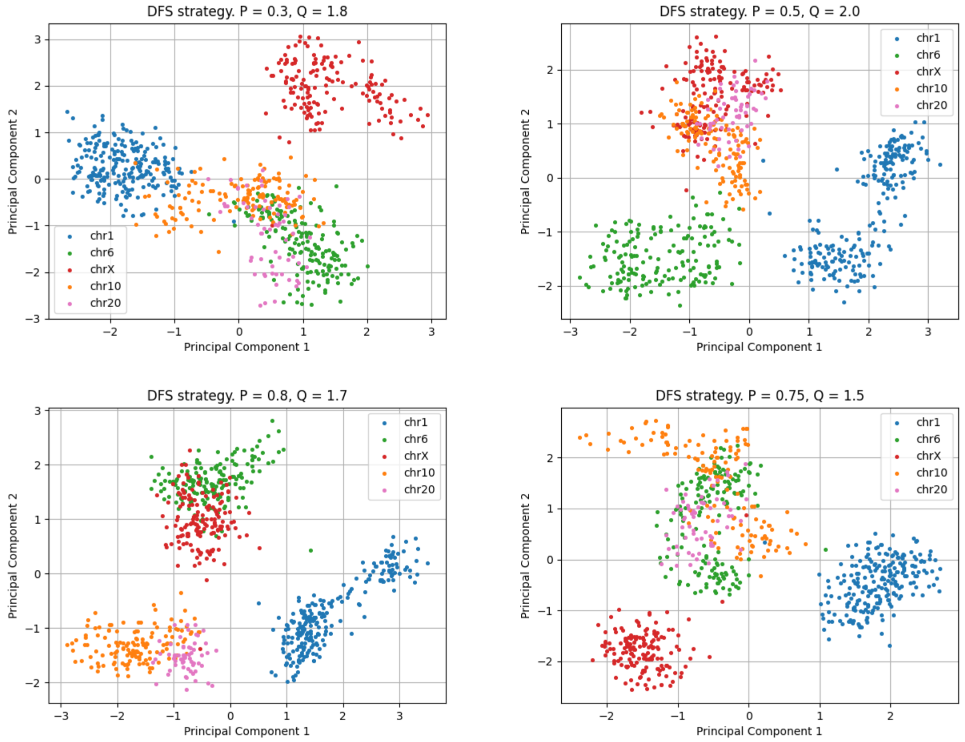

Figure 9: PCA on Node2Vec embeddings with Depth-first search (DFS) strategy: $p < q$. The parameters $p = 0.8$ and $q = 1.7$ seem to identify both translocations 10-20 and 6-X.

graph, much like the breadth-first search strategy. This exploration can yield embeddings that reflect the broader, more global properties of the graph. Conversely, a lower q value places a stronger emphasis on the exploration of highly connected regions, effectively emphasizing the local structure, similar to the depth-first search strategy. The modification of parameter q can have an impact on the balance between capturing local and global graph characteristics within PCA visualizations. With higher q values, we may find our PCA plots highlighting global structural aspects, unveiling broader relationships between nodes, resembling BFS. Conversely, lower q values may emphasize local, densely connected patterns in the visualizations, akin to the localized nature of DFS. In this specific case, the parameters $p = 0.8$ and $q = 1.7$ appear to identify both translocations 10-20 and 6-X. However, this appears to be an erroneous identification. Upon increasing the number of walks and the length of walks, it has been observed that translocation 10-20 persists while the identification of 6-X disappears (see Figure 10).

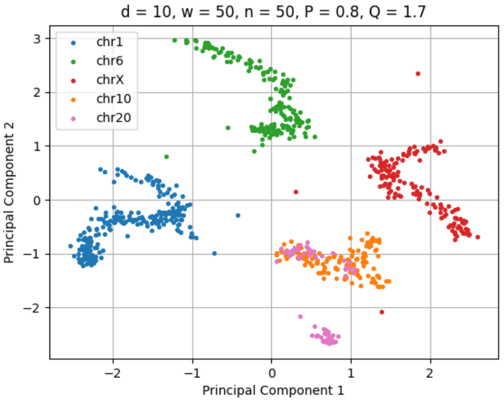

Figure 10: By fine-tuning the parameters of the Node2Vec algorithm to track both translocations and increasing the number of walks and walk length, we succeeded in consistently detecting the 10-20 translocation. However, the 6-X translocation disappeared from our observations under these conditions.

6 Conclusions

In conclusion, this study resulted in the development of a compact library designed to identify chromosome translocations using Hi-C data, which is interpreted as an adjacency matrix. The research focused on a tumor cell with experimentally verified chromosome translocations involving chromosome pairs 10-20 and 6-X. We attempted to detect these translocations using the Node2Vec algorithm, complemented by a principal component analysis.

The identification of the 10-20 translocation occurred rapidly. To detect 6-X translocations, we experimented with various parameters of the Node2Vec algorithm and explored different approaches to construct block graphs to isolate node indices related to the affected chromosomes.

Out of all the cases examined, only one yielded the observation of both anticipated translocations. This outcome was achieved by adjusting the parameters in the following manner $d = 10$, $p = 0.8$, $q = 1.7$, $w_l = 10$, $n_w = 10$.

After obtaining these results, we made an effort to enhance the number of walks and the walk length in an attempt to capture more intricate graph features. Unfortunately, this adjustment led to the loss of the 6-X translocation while retaining the 10-20 translocation. Consequently, it can be concluded that this technique performs admirably in detecting a specific type of translocation: indeed, the 6-X translocation, as it emerges in the Hi-C matrix, can be described not as an exchange of material between chromosomes (i.e., some nodes of chr10 have been transferred to chr20 and vice versa) but rather as a sharing of nodes (i.e., some nodes of chr6 belong to both chr6 and chrX, and vice versa). In this sense, working with an algorithm setup aimed at identifying communities in the strict sense may not be the best solution in this case.

However, this methodology can prove to be an efficient and cost-effective computational method for observing translocations when there is an actual exchange of genetic material between chromosomes within a cell.

References

[1] Erez Lieberman-Aiden et al. “Comprehensive mapping of long-range interactions reveals folding principles of the human genome”. In: *science* 326.5950 (2009), pp. 289–293.

[2] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks”. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining.* 2016, pp. 855–864.

[3] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[4] *Node2VecHiC library GitHub site. Author: Keivan Amini.* `https://github.com/keivan-amini/Node2VecHiC`.

[5] Fang Hu et al. “Community detection in complex networks using Node2vec with spectral clustering”. In: *Physica A: Statistical Mechanics and its Applications* 545 (2020), p. 123633.

[6] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[7] *Node2Vec library GitHub site. Author: Elior Cohen.* `https://github.com/eliorc/node2vec`.

[8] Gareth James et al. *An introduction to statistical learning.* Vol. 112. Springer, 2013.

[9] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction.* Vol. 2. Springer, 2009.

A Appendix

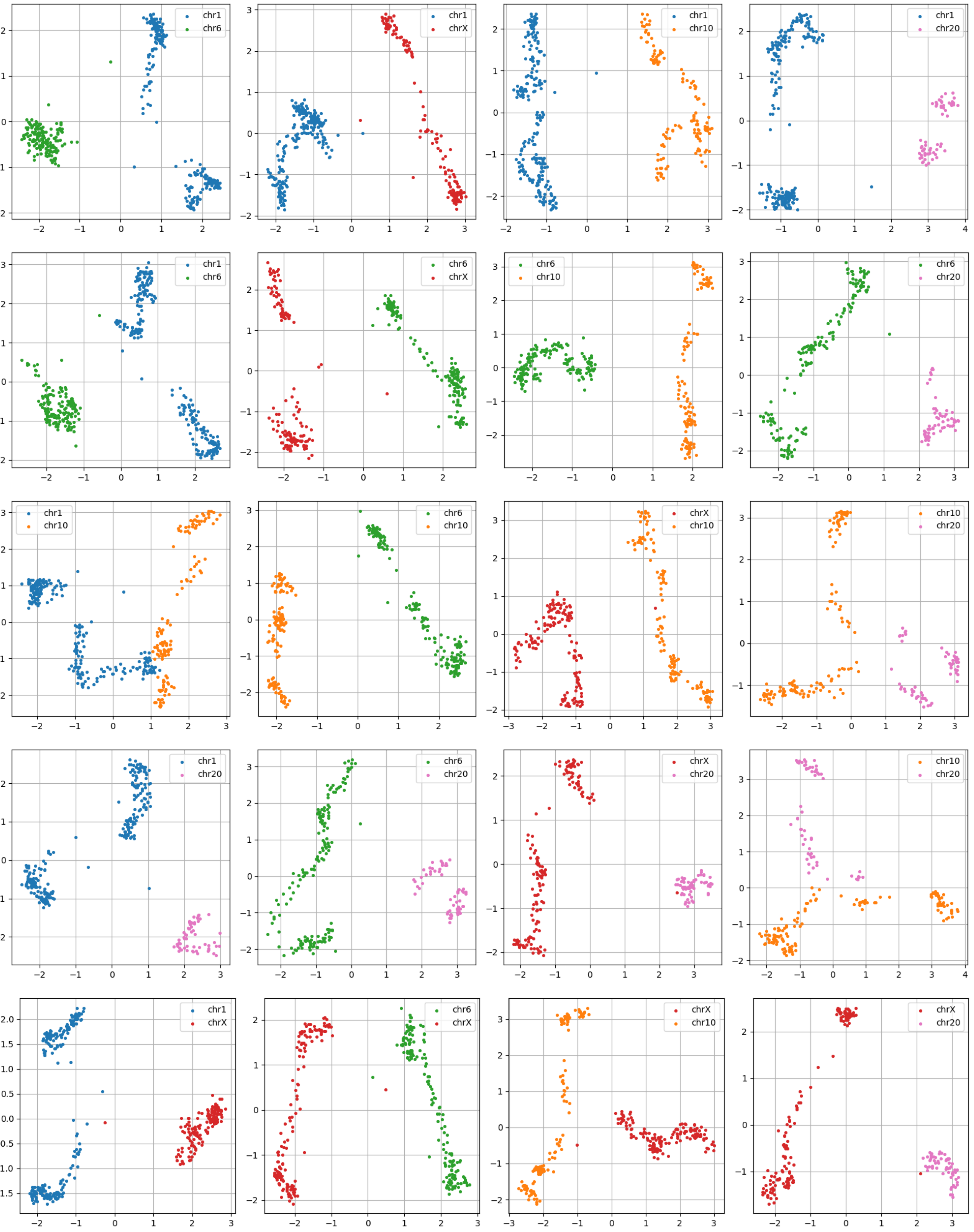

Figure 11: PCA on Node2Vec embeddings with all possible chromosomes couple of the G_c graph. Unfortunately, this technique fails to detect chromosome transmutations. In no case are the experimentally confirmed translocations associated with chromosomes 10-20 and 6-X observed: node isolation must be discarded in this context.