MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach

Léo Pio-Lopez, Alberto Valdeolivas, Laurent Tichit, Élisabeth Remy, Anaïs Baudot

scientific reports

Check for updates

OPEN

MultiVERSE: a multiplex and multiplex-heterogeneous network embedding approach

Léo Pio-Lopez[1]✉, Alberto Valdeolivas[2], Laurent Tichit[1], Élisabeth Remy[1] & Anaïs Baudot[3,4]

Network embedding approaches are gaining momentum to analyse a large variety of networks. Indeed, these approaches have demonstrated their effectiveness in tasks such as community detection, node classification, and link prediction. However, very few network embedding methods have been specifically designed to handle multiplex networks, i.e. networks composed of different layers sharing the same set of nodes but having different types of edges. Moreover, to our knowledge, existing approaches cannot embed multiple nodes from multiplex-heterogeneous networks, i.e. networks composed of several multiplex networks containing both different types of nodes and edges. In this study, we propose MultiVERSE, an extension of the VERSE framework using Random Walks with Restart on Multiplex (RWR-M) and Multiplex-Heterogeneous (RWR-MH) networks. MultiVERSE is a fast and scalable method to learn node embeddings from multiplex and multiplex-heterogeneous networks. We evaluate MultiVERSE on several biological and social networks and demonstrate its performance. MultiVERSE indeed outperforms most of the other methods in the tasks of link prediction and network reconstruction for multiplex network embedding, and is also efficient in link prediction for multiplex-heterogeneous network embedding. Finally, we apply MultiVERSE to study rare disease-gene associations using link prediction and clustering. MultiVERSE is freely available on github at https://github.com/Lpiol/MultiVERSE.

Networks are powerful representations to describe, visualize, and analyse complex systems in many domains. Recently, machine learning techniques started to be used on networks, but these techniques have been developed for vector data and cannot be directly applied. A major challenge thus pertains to the encoding of high-dimensional graph-based data into a feature vector. Network embedding (also known as graph representation learning) provides a solution to this challenge and allows opening the complete machine learning toolbox for network analysis.

The high efficiency of network embedding approaches has been demonstrated in a wide range of applications such as community detection, node classification, or link prediction. Moreover, network embedding approaches can exploit very large graphs, with millions of nodes[1]. Thus, with the explosion of big data, network embeddings have been used to study many different networks, such as social[2], neuronal[3] and molecular networks[4].

So far, network embedding approaches have been mainly applied to monoplex networks (i.e. single networks composed of one type of nodes and edges)[1,5,6]. Current technological advances however generate a large spectrum of data, which form large heterogeneous datasets. Single monoplex networks are not suited to represent such diversity. Therefore, multi-layer networks, including multiplex[7] and multiplex-heterogeneous[8] networks have been proposed to handle these richer sets of relationships.

Multiplex networks are composed of several layers, each layer being a monoplex network. All the layers share the same set of nodes, but their edges belong to different categories (Fig. 1A). Multiplex representation is pertinent to depict the diversity of interactions between the same nodes. For instance, in a molecular multiplex network, the different layers could represent physical interactions between proteins, their belonging to the same molecular complexes or the correlation of expression of the genes across different tissues. Analogously, in social multiplex networks, a person can belong to different layers describing different types of relationships, such as friendships or common interests.

A heterogeneous network is a multi-layer network in which each layer is a monoplex network with its specific type of nodes and edges (Fig. 1B). The two monoplex networks are connected by bipartite interactions, *i.e.* edges linking the different types of nodes belonging to the two monoplex networks. Such heterogeneous networks have

[1]Aix Marseille Univ, CNRS, Centrale Marseille I2M, Marseille, France. [2]Heidelberg University, Institute for Computational Biomedicine, Heidelberg, Germany. [3]Aix Marseille Univ, INSERM, CNRS, MMG, Marseille, France. [4]Barcelona Supercomputing Center, Barcelona, Spain. ✉email: leo.pio.lopez@gmail.com

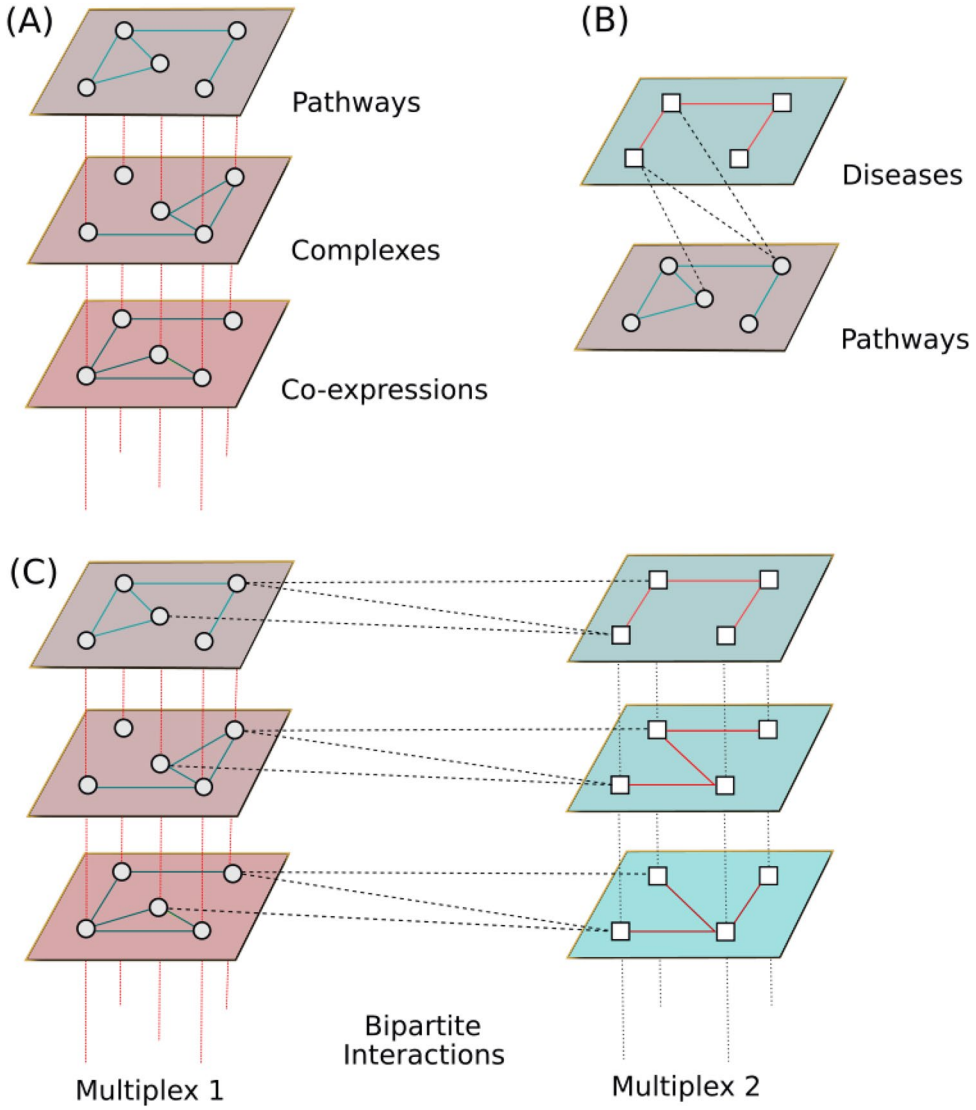

Figure 1. Illustrations of the different types of networks. (**A**) A multiplex network. The different layers share the same set of nodes but different types of edges. (**B**) A heterogeneous network. The two networks are composed of different types of nodes and edges, connected by bipartite interactions (black dashed lines). (**C**) A multiplex-heterogeneous network composed of two multiplex networks. The multiplex networks are connected by bipartite interactions (dashed lines). For the sake of simplicity, the figure does not represent all the possible bipartite interactions (each layer of a given multiplex is in reality linked with every layer of the other multiplex).

been studied in different research fields. For example, in network medicine, a drug-protein target heterogeneous network has been constructed with a drug-drug similarity monoplex network, a protein-protein interaction monoplex network and bipartite interactions between drugs and their target proteins[9]. In social science, citation networks are constructed with author-author and document-document monoplex networks connected by author-documents bipartite interactions, as in[10].

A multiplex-heterogeneous network is a combination of heterogeneous and multiplex networks by connecting several multiplex networks through bipartite interactions (Fig. 1C). The multiplex-heterogeneous structure is expected to provide a richer view on biological[8], social[11] or other real-world systems describing complex relations among different components.

Recently, different studies proposed embedding approaches for multiplex networks[11–14] and heterogeneous networks[15,16]. A recent method uses multiplex-heterogeneous information to embed one category of nodes[17]. However, to our knowledge, no embedding methods are specifically dedicated to the embedding of nodes of different types from multiplex-heterogeneous networks. In this paper, we present MultiVERSE, a fast, scalable and versatile embedding approach to learn node embeddings on multiplex and multiplex-heterogeneous networks. MultiVERSE is based on the VERSE framework[18], and coupled with Random Walks with Restart on Multiplex (RWR-M) and on Multiplex-heterogeneous (RWR-MH) networks[8]. Our contributions are the following:

- We propose an evaluation protocol in order to evaluate multiplex network embedding. It is based on 7 datasets in 4 disciplines (biological, neuronal, co-authorship and social networks), 6 embedding methods (and 4 additional link prediction heuristics), and two tasks: link prediction and a new protocol approach based on network reconstruction.
- We demonstrate the higher performance of MultiVERSE over state-of-the-art network embedding methods in the tasks of link prediction and network reconstruction for multiplex network embedding.
- We propose, to our knowledge, the first multiplex-heterogeneous network embedding method (with an embedding of the different types of nodes).
- We propose a method to evaluate multiplex-heterogeneous network embedding on link prediction. We demonstrate the effectiveness of MultiVERSE on this task on two biological multiplex-heterogeneous networks.
- We present a biological application of MultiVERSE for the study of gene-disease associations using link prediction and clustering.

Related work in network embedding

Network embedding relies on two key components: a similarity measure between pairs of nodes in the original network and a learning algorithm. Given a network and a similarity measure, the aim of network embedding is to learn vector representations of the nodes in a lower dimension space, while preserving as much as possible the similarity. In the next sections we will present the state-of-the-art of monoplex, multiplex and multiplex-heterogeneous network embedding.

Monoplex network embedding. Many network embedding methods have been recently developed to study a large variety of networks, from biological to social ones. The classical method deepwalk[5] inspired a series of methods such as node2vec[6] and LINE (for Large-scale Information Network Embedding)[19]. Deepwalk uses truncated random walks to compute the node similarity in the network. Then, a combination of the skip-gram learning algorithm[20] and hierarchical softmax[21] is used to learn the graph representations. Skip-gram is a model based on natural language processing. It intends to maximize the probability of co-occurrence of nodes within a walk, focusing on a window, i.e. a section of the path around the node. Node2vec[6] upgrades deepwalk by introducing negative sampling during the learning phase[22]. Moreover, node2vec allows biasing the random walks towards depth or breadth-first random walks, in order to tune the exploration of the search space. LINE[19] follows a different approach to optimize the embedding: it computes the node similarity using an adjacency-based proximity measure in association with negative sampling. Other embedding methods are based on matrix-factorization, such as GraRep[23] or HOPE[24]. It has been shown that random-walk based methods for network embedding can be expressed in terms of matrix-factorization[25]. Another class of methods are based on neural networks such as GraphSAGE[26], graph convolutional networks (GCN)[27] or graph auto-encoders (GAE/VGAE)[28].

These embedding methods have been applied to link prediction or node labelling tasks. Their performance rely upon multiple criteria such as the size of the network, its density, the embedding dimension or the evaluation metrics[29]. Overall, they have been designed to handle monoplex networks. However, we now have access to a richer representation of complex systems as multiplex networks, and some recent methods have explored the embedding of such multiplex networks.

Multiplex network embedding. The most straightforward approach to deal with multiplex networks is to merge the different layers into a monoplex network[30]. However, this merging creates a new network with its own topology, and loses the topological features of the individual layers. This new topology is logically biased towards the initial topology of the denser layers[31]. Different network embedding methods have been introduced in order to avoid merging multiplex network layers and take advantage of the multiplex structure[11–14]. Overall, these approaches are based on truncated random walks to compute the similarity in the multiplex network. Ohmnet[13] relies on node2vec[6] and requires the definition of a hierarchy of layers to model dependencies between them. But usually, this layer hierarchy information is not known or easy to establish, particularly for multiplex networks such as social or molecular networks. The Scalable Multiplex Network Embedding (MNE) method[12] is also based on node2vec[6]. For each network node, it extracts one high-dimensional common embedding shared across all the layers of the multiplex network. In addition, MNE computes a lower-dimensional embedding for every node in each layer of the multiplex network. Multi-node2vec[14] is another method based on node2vec that constructs the multiplex embedding with the random walks jumping from one layer to another. Multi-Net[11] also proposes a random walks procedure in the multiplex network, inspired from[32]. Similarly to multi-node2vec, the random walks can jump from one layer to another. Multi-Net learns the embeddings using stochastic gradient descent. The performances of Ohmnet[13], Multi-net[11] and MNE[12] have been compared in the context of network reconstruction[11]. In this task, the aim is to reconstruct one layer of the multiplex network from the embeddings of the other layers. The results show better performances for Multi-net on a set of social and biological multiplex networks[11].

Multiplex-heterogeneous network embedding. Some methods can perform the embedding of heterogeneous networks[15,16]. A famous approach is metapath2vec[15]. It extends skip-gram to learn node embeddings for heterogeneous networks using meta-paths, which are predefined composite relations between different types of nodes. For instance, in the context of a drug-protein target heterogeneous network, the meta-path drug-protein target-drug in the network could bias the random walks to extract the information related to drug combinations.

Nevertheless, to our knowledge, no approach is specifically dedicated to the embedding of different types of nodes from multiplex-heterogeneous networks. In the next section, we present formally MultiVERSE, a new

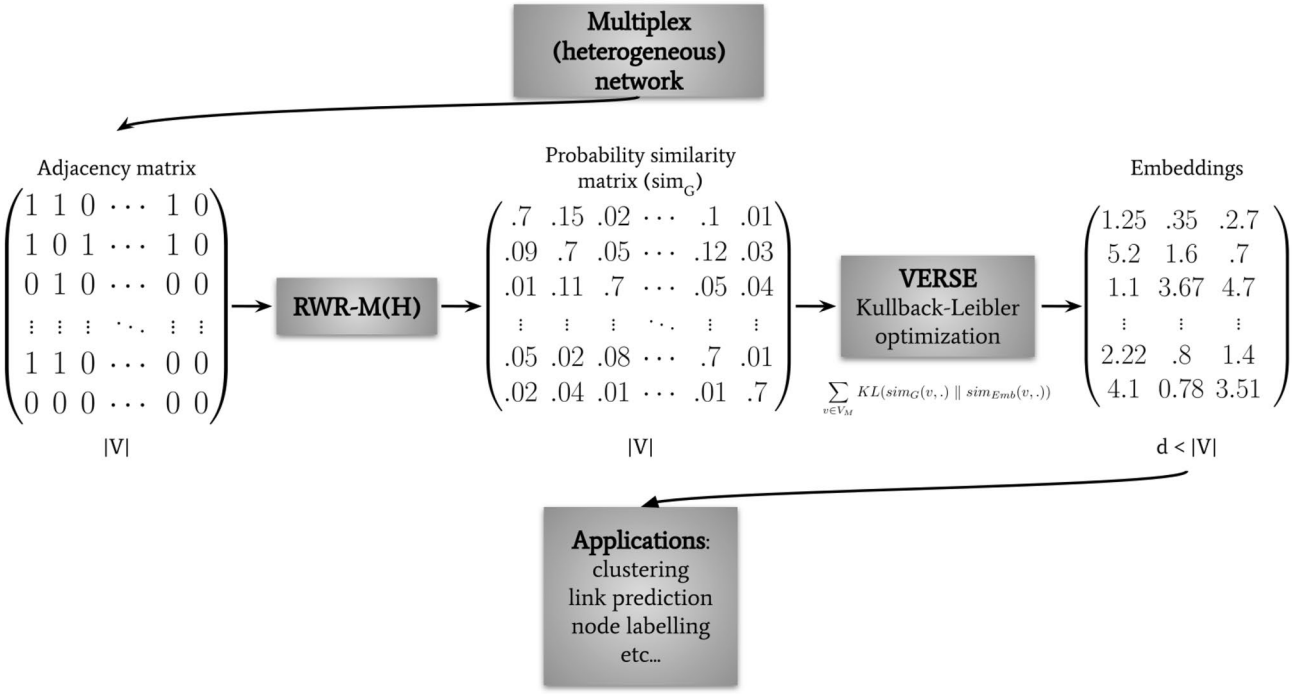

Figure 2. Overview of the MultiVERSE pipeline. Starting from a multiplex-heterogeneous network, we represent its structure through an adjacency matrix (size $|V| \times |V|$); we then compute a similarity matrix using Random Walk with Restart algorithm, and apply an optimized version of the VERSE algorithm to compute the embeddings. The resulting matrix of embeddings will be used for the applications.

method for multiplex and multiplex-heterogeneous network embedding relying on VERSE[18] and coupled with Random Walks with Restart extended to Multiplex (RWR-M) and Multiplex-Heterogeneous graphs (RWR-MH)[8].

MultiVERSE

In this section, we present the key components of MultiVERSE: the VERSE general framework, the learning objective, and our particular implementation with Random Walk with Restart for Multiplex networks (RWR-M) and Random Walk with Restart for Multiplex-Heterogeneous networks (RWR-MH) (Fig. 2). We finally describe the MultiVERSE algorithm.

VERSE: a general framework for network embedding. The aim of VERSE network embedding is to learn a low-dimensional nonlinear representation w_i of the nodes v_i to a d-dimensional continuous vector, where $d < n$, using Kullback-Leibler optimization[18]. We denote d the dimension of the embedding space, and n the dimension of the adjacency matrix. VERSE was originally developed for the embedding of monoplex networks[18]. The VERSE framework is nevertheless general and versatile enough to be expanded to multiplex and multiplex-heterogeneous networks.

Similarity distributions. Consider an undirected graph $G = (V, E)$ with $V = \{v_i, i = 1, \ldots, n\}$ the set of nodes ($|V| = n$), and $E \subseteq V \times V$ the set of edges, and $sim_G : V \times V \to \mathbb{R}$ a given similarity measure on G such that

$$\forall v \in V, \sum_{u \in V} sim_G(v, u) = 1\,. \tag{1}$$

Hence, the similarity for any node v is expressed as a probability distribution $sim_G(v, .)$.

We note w_i the vector representation of node i in the embedding space (W is a $(n \times d)$-matrix). The (non-normalized) similarity between two nodes embeddings w_u and w_v is defined as the dot product $w_u \cdot w_v^T$. Using the softmax function, we obtain the normalized similarity distribution in the embedding or vector space:

$$sim_{Emb}(v, .) = \frac{exp(w_v \cdot w^T)}{\sum_{i=1}^{n} exp(w_v \cdot w_i)}\,. \tag{2}$$

Finally, the output of any network embedding method is a matrix of embeddings W such as, $\forall v \in V$, $sim_{Emb}(v, .) \approx sim_G(v, .)$. This requires a learning phase, which is described in the next section.

Learning objective. This step updates the embeddings at each iteration in order to project sim_G into the embedding space leading to the preservation of the topological structure of the graph. In the framework of VERSE,

as sim_{Emb} and sim_G are both probability distributions, this optimization phase aims to minimize the Kullback-Leibler divergence (KL-divergence) between these two similarities:

$$\sum_{v \in V_M} KL(sim_G(v,.) \parallel sim_{Emb}(v,.)) \tag{3}$$

We can keep only the parts related to sim_{Emb} as it is the target to optimize and sim_G is constant. This leads to the following objective function:

$$\mathcal{L} = -\sum_{v \in V_M} sim_G(v,.)\, log(sim_{Emb}(v,.)) \tag{4}$$

sim_{Emb} is defined as a softmax function and needs to be normalized over all the nodes of the graph at each iteration, which is computationally heavy. Therefore, following the VERSE algorithm[18], we used Noise Contrastive Estimation (NCE) to compute this objective function[33,34]. NCE trains a binary classifier to distinguish node samples coming from the distribution of similarity in the graph sim_G and those generated by a noise distribution Q. We define D as the random variable representing the classes, $D = 0$ for a node if it has been drawn from the noise distribution Q or $D = 1$ if it has been drawn from the empirical distribution and $\mathbb{E}$ is the expected value. With u a node drawn from $\mathcal{P}$ and v drawn from $sim_G(u,.)$, with NCE we draw $s < n$ negative samples v_{neg} from $Q(u)$.

In this framework, the objective function becomes the negative log-likelihood that we want to minimize via logistic regression:

$$\begin{aligned} \mathcal{L}_{NCE} = \sum_{\substack{u \sim \mathcal{P} \\ v \sim sim_G(u,.)}} \Big[& log P_W(D = 1 \mid sim_{Emb}(u,v)) \\ & + s.\mathbb{E}_{v_{neg} \sim Q(u)} log P_W(D = 0 \mid sim_{Emb}(u,\widetilde{v})) \Big] \end{aligned} \tag{5}$$

where P_W is computed as the sigmoid ($\sigma(x) = (1 + e^{-x})^{-1}$) of the dot product of the embeddings w_u and w_v, and $sim_{Emb}(u,.)$ is computed without normalization. It has been proven that the derivative of NCE converges to gradient of cross-entropy when s increases, but in practice small values work well[34]. Therefore, we are minimizing the KL-divergence from sim_G.

Overall, VERSE is a general framework for network embedding with the only constraint that sim_G must be defined as a probability distribution. In this work, we computed sim_G using Random Walks with Restart on Multiplex (RWR-M) and Random Walks with Restart on Multiplex-Heterogeneous (RWR-MH) networks[8]. We describe this particular implementation in the next section.

Random walk with restart on multiplex and multiplex-heterogeneous networks. *Random walk (RW) and random walk with restart (RWR).* Let us consider a finite graph, $G = (V, E)$, with adjacency matrix A. In a classical RW, an imaginary particle starts from a given initial node, v_0. Then, the particle moves to a randomly selected neighbour of v_0 with a probability defined by its degree. We can define $p_t(v)$ as the probability for the random walk to be at node v at time t. Therefore, the evolution of the probability distribution, $\mathbf{p}_t = (p_t(v))_{v \in V}$, can be described as follows:

$$\mathbf{p}_{t+1}^T = M\mathbf{p}_t^T \tag{6}$$

where M denotes a transition matrix that is the column normalization of A. The stationary distribution of Eq. (6) represents the probability for the particle to be located at a specific node when times tends to infinity[35].

Random Walk with Restart (RWR) additionally allows the particle to jump back to the initial node(s), known as seed(s), with a probability $r \in (0, 1)$ at each step. In this case, the stationary distribution can be interpreted as a measure of the proximity between the seed(s) and all the other nodes in the graph. We can formally define RWR by including the restart probability in Eq. (6):

$$\mathbf{p}_{t+1}^T = (1 - r)M\mathbf{p}_t^T + r\mathbf{p}_0^T \tag{7}$$

The vector $\mathbf{p}_0$ is the initial probability distribution. Therefore, in $\mathbf{p}_0$, only the seed(s) have values different from zero. Equation (7) can be solved in a iterative way[8].

In our previous work, we expanded the Random Walk with Restart algorithm to Multiplex (RWR-M) and Multiplex-Heterogeneous networks (RWR-MH)[8]. Below, we show how the output of RWR-M and RWR-MH can easily be adapted to produce sim_G, the required input for the VERSE framework.

Random walk with restart on multiplex networks (RWR-M). We define a multiplex graph as a set of L undirected graphs, termed layers, which share the same set of n nodes[7,36]. The different layers, $\alpha = 1, \ldots, L$, are defined by their respective $n \times n$ adjacency matrices, $A^{[\alpha]} = (A^{[\alpha]}(i,j))_{i,j=1,\ldots,n}$. $A^{[\alpha]}(i,j) = 1$ if node i and node j are connected on layer α, and 0 otherwise[37]. We do not take into account potential self-interactions and therefore set $A^{[\alpha]}(i,i) = 0\ \forall\ i = 1, \ldots, n$. In addition, we consider that v_i^α represents the node i in layer α.

Thus, we can represent a multiplex graph by its adjacency matrix:

$$\mathbf{A} = A^{[1]}, \ldots, A^{[L]} \tag{8}$$

and define it as $G_M = (V_M, E_M)$, where:

$$V_M = \left\{ v_i^\alpha,\ i = 1, \ldots, n,\ \alpha = 1, \ldots, L \right\},$$
$$E_M = \left\{ (v_i^\alpha, v_j^\alpha),\ i,j = 1, \ldots, n,\ \alpha = 1, \ldots, L,\ A^{[\alpha]}(i,j) \neq 0 \right\} \bigcup \left\{ (v_i^\alpha, v_i^\beta),\ i = 1, \ldots, n,\ \alpha \neq \beta \right\}.$$

RWR-M should ideally explore in parallel all the layers of a multiplex graph to capture as much topological information as possible. Therefore, a particle located in a given node, v_i^α, may be able to either walk to any of its neighbours within the layer α or to jump to its counterpart node in another layer, v_i^β with $\beta \neq \alpha$[38]. Additionally, the particle can restart in the seed node(s) on any layer of the multiplex graph. In order to match these requirements, we previously defined a multiplex transition matrix and expanded the restart probability vector, allowing us to apply Eq. (6) on multiplex graphs[8].

In this study, we independently run the RWR-M algorithm n times, using each time a different node as seed. As a result, we obtain a $n \times n$ matrix in which each column describes the probability of finding the particle in every network node when the steady state is reached. We use this probability distribution as a measure of similarity between a given node and all the other nodes of the multiplex graph. Hence, we have $\sum_{u \in V_M} sim_G(v,u) = 1\ \forall\, v \in V_M$, therefore fulfilling the requirements of the VERSE input. We set the RWR-M parameters to the same values used in our original study ($r = 0.7, \boldsymbol{\tau} = (1/L, 1/L, \ldots, 1/L), \delta = 0.5$)[8].

Random walk with restart on multiplex-heterogeneous networks (RWR-MH). A heterogeneous graph is composed of two graphs with different types of nodes and edges. In addition, it also contains a bipartite graph in order to link the nodes of different type (bipartite edges)[39]. In our previous study[8], we described how to extend the RWR to a graph which is both multiplex and heterogeneous. However, this study considered only one multiplex graph in the multiplex-heterogeneous graph. For the present work, we additionally expanded RWR-MH to a complete multiplex-heterogeneous graph, i.e. both components of the heterogeneous graph can be multiplex (Fig. 1C), based on the work of[40]. Let us consider a L-layers multiplex graph, $G_M = (V_M, E_M)$, with $n \times L$ nodes, $V_M = \left\{ v_i^\alpha,\ i = 1, \ldots, n,\ \alpha = 1, \ldots, L \right\}$. We also define a second L-layers multiplex graph, with $m \times L$ nodes, $U_M = \left\{ u_j^\alpha,\ j = 1, \ldots, m,\ \alpha = 1, \ldots, L \right\}$. We additionally need a bipartite graph $G_B = (V_M \cup U_M, E_B)$ with $E_B \subseteq V_M \times U_M$. The edges of the bipartite graph only connect pairs of nodes from the different sets of nodes, V_M and U_M. It is to note that the bipartite edges should link nodes with every layer of the multiplex graphs. We therefore need L identical bipartite graphs, $G_B^{[\alpha]} = (V_M \cup U_M, E_B^{[\alpha]})$ to define the multiplex-heterogeneous graph. We can then describe a multiplex-heterogeneous graph, $G_{MH} = (V_{MH}, E_{MH})$, as:

$$V_{MH} = \{V_M \cup U_M\}$$
$$E_{MH} = \left\{ \cup_{\alpha=1,\ldots,L} E_B^{[\alpha]} \cup E_{V_M} \cup E_{U_M} \right\}$$

In the RWR-MH algorithm, the particle should be allowed to move in any of the multiplex graphs as described in the RWR-M section. In addition, it may be able to jump from a node in one multiplex graph to the other multiplex graph following a bipartite edge. We also have to bear in mind that the particle could now restart in different types of node(s), i.e. we can have seed(s) of different category (see Fig. 1C). We accordingly defined a multiplex-heterogeneous transition matrix and expanded the restart probability vector. This gave us the opportunity to extent and apply Eq. (6) on multiplex-heterogeneous graphs[8,40].

In the context of MultiVERSE, we independently run the RWR-MH algorithm $n + m$ times. In each execution, we select a different seed node until all the nodes from both multiplex graphs have been used as individual seeds. As a result, we can define a node-to-node similarity matrix matching VERSE input criteria, i.e $\sum_{u \in V_{MH}} sim_G(v,u) = 1\ \forall\, v \in V_{EM}$. We set the RWR-MH parameters to the same values used in the original study ($r = 0.7, \boldsymbol{\tau} = (1/L, 1/L, \ldots, 1/L), \delta = 0.5, \lambda = 0.5, \eta = 0.5$)[8].

MultiVERSE algorithm. Algorithm 1 presents the pseudo-code of MultiVERSE based on RWR on multiplex and multiplex-heterogeneous networks[8] and Kullback-Leibler optimization from the VERSE algorithm[18].

Our implementation of VERSE with NCE is slightly different from the original. We perform first the RWR-M or RWR-MH for all the nodes of the network in order to obtain the similarity distribution sim_{G_M}. The output of this step is the probability matrix $\overline{\mathbf{p}}$, where $\overline{\mathbf{p}}_u$ is the probability vector representing the similarities between u and all the other nodes. The matrix of the embedded representation of the nodes, W, is randomly initialized. For each iteration, from one node u sampled randomly from a uniform distribution $\mathcal{U}$, we truncate the probability vector $\overline{\mathbf{p}}_u$. We keep the N_{max} highest probabilities because the shape of the distribution of probabilities falls very fast to very low probabilities. Doing so, we can speed up the calculation and reduce memory constraints by filtering out the lowest probabilities and by reducing the size of the similarity matrix. We normalize this resulting probability vector $\dot{\mathbf{p}}_u$, and sample one node v according to its probability in $\overline{\mathbf{p}}_u$. We set empirically the parameter $N_{max} = 300$ for networks with more than 5000 nodes. For smaller networks, we set this parameter to 10% − 20% of the number of nodes of the network, depending on the shape of the distribution. These choices of N_{max} have been done for memory and quality of embeddings reasons. Indeed, these values are sufficient to obtain high quality embeddings, and avoid storing and manipulating the whole output of RWR-M(H), which is a $n \times n$ matrix. We store with this truncated sampling strategy a $n \times N_max$ matrix. These two steps (lines 6 and 7) were not in the

original VERSE. We parallelized the repeat loop (line 4) and added a parallelized for loop after line 5 in order to run the code from line 6 to 12 in parallel P times. In our simulations, we set $P = 100$.

Algorithm 1 MultiVERSE algorithm

1: **Input:** a multiplex graph, N_{max}, s
2: $W \leftarrow \mathcal{N}(0,1)$
3: $\overline{\mathbf{p}} \leftarrow$ RWR-M(H)(G_M)
4: **repeat**
5: $\quad u \sim \mathcal{U}$
6: $\quad \dot{\mathbf{p}}_u = \text{Normalize}(\overline{\mathbf{p}}_u(1,...,N_{max}))$
7: $\quad v_{pos} \sim \dot{\mathbf{p}}_u$
8: $\quad W_u, W_{v_{pos}} \leftarrow Update(u,\ v_{pos},\ 1,\ bias_{pos})$
9: $\quad$ **for** i=1, ... , s **do**
10: $\quad\quad v_{neg} \sim \mathcal{Q}(u)$
11: $\quad\quad W_u, W_{v_{neg}} \leftarrow Update(u,\ v_{neg},\ 0,\ bias_{neg})$
12: $\quad$ **end for**
13: **until** Maximum step reached

Algorithm 2 Update

1: **Input:** u, v, D, $bias$, lr
2: $g \leftarrow [D - \sigma(W_u \cdot W_v - bias)] * lr$
3: $W_u \leftarrow g \cdot W_v$

Then, we update W_u and W_v according to algorithm 2 by reducing their distances in the embedding space. We added the bias for NCE: $bias_{pos} = log(N)$ and $bias_{neg} = log(N/s)$.

Then, s negative nodes are sampled from $\mathcal{Q}(u)$ and we update the corresponding embeddings by increasing their distances in the embedding space. The parameter s has been set to s=10 for networks with a number of nodes superior to 5000 and to $s = 3$ as in the VERSE original algorithm for smaller networks. The precision of the NCE depends on the parameter s, and small values work well in practice[34]. The update can also be seen as the training part with lr as the learning rate of the binary classifier of the NCE estimation as described in Eq. (5). The whole process is repeated until the maximum steps are reached.

Regarding computational time, it depends mainly on the available number of cores and number of nodes ((as RWR has a time complexity of $\mathcal{O}(n^2)$)=. On a i7-6820HQ CPU @2.70GHz with 8 cores and 48 Gb of RAM, the whole computation of MultiVERSE for the molecular multiplex network (see next section) with $d = 128$ and $s = 10$ takes 45 minutes.

MultiVERSE is freely available on github at https://github.com/Lpiol/MultiVERSE.

Evaluation protocol

We propose a benchmark to compare the performance of MultiVERSE and other embedding methods for multiplex and multiplex-heterogeneous networks. The performances are evaluated through link prediction for both multiplex and multiplex-heterogeneous networks, and with network reconstruction for multiplex networks.

Evaluation of multiplex network embedding. In the next sections, we describe the datasets, the evaluation tasks and the methods used for evaluations.

Multiplex network datasets. We used 7 multiplex networks (2 molecular, 1 disease, 1 neuronal, 1 co-authorship and 2 social networks) to evaluate the different approaches of multiplex network embedding. The networks CKM, LAZEGA, C.ELE, ARXIV, and HOMO have been extracted from the CoMuNe lab database https://comunelab.fbk.eu/data.php. We constructed the other two networks, DIS and MOL. A description of each of these multiplex networks follows. The number of nodes and edges of the different layers are detailed in Table 1.

- *CKM physician innovation (CKM)* is a multiplex network describing how physicians in four towns in Illinois used the new drug tetracycline[41]. It is composed of 3 layers corresponding to three questions asked to the physicians: i) to whom do you usually turn when you need information or advice about questions of therapy? ii) who are the three or four physicians with whom you most often find yourself discussing cases or therapy in the course of an ordinary week – last week for instance? iii) would you tell me the first names of your three friends whom you see most often socially?
- *Lazega network (LAZEGA)* is a multiplex social network composed of 3 layers based on co-working, friendship and advice between partners and associates of a corporate law partnership[42].

Dataset	Layers	Nodes	Edges
CKM	1	215	449
	2	231	498
	3	228	423
LAZEGA	1	71	717
	2	69	399
	3	71	726
C.ELE	1	253	514
	2	260	888
	3	278	1703
ARXIV	1	1558	3013
	2	5058	14,387
	3	2826	6074
	4	1572	4423
	5	3328	7308
	6	1866	4420
	7	1246	1947
	8	4614	11,517
HOMO	1	12,,345	48,528
	2	14,770	83,414
	3	1626	1953
	4	5680	18,381
DIS	1	3891	117,527
	2	4155	101,104
	3	434	3137
MOL	1	14,704	122,211
	2	7926	194,500
	3	8537	63,561

Table 1. Description of the 7 multiplex networks used for the evaluation protocol.

- *Caenorabidis Elegans connectome (C.ELE)* is a neuronal multiplex network composed of 3 layers corresponding to different synaptic junctions[43,44]: electrical, chemical poladic and chemical monadic.
- *ArXiv network (ARXIV)* is composed of 8 layers corresponding to different ArXiv categories. The dataset has been restricted to papers with 'networks' in the title or abstract, up to May 2014[45]. The original data from the CoMuNe Lab database is divided in 13 layers. We extracted the 8 layers (1-2-3-5-6-8-11-12) containing more than 1000 edges.
- *Homo sapiens network (HOMO)* is composed of 4 layers extracted from the original network on CoMuNe Lab[44], keeping physical association, direct interaction, association and co-localization layers. The data are initially extracted from BioGRID[46]
- *Disease multiplex network (DIS)* has been constructed, composed of 3 layers: i) A disease-disease network based on a projection of a disease-drug network from the Comparative Toxicogenomics Database (CTD)[47] extracted from BioSNAP[48]. In this network, an edge between two diseases is created if the Jaccard Index between the neighborhoods of the two nodes in the original bipartite network is superior to 0.4. Two diseases are thereby linked if they share a similar set of drugs. This projection has been done using NetworkX[49]. ii) A disease-disease network where the edges are based on shared symptoms. The network has been constructed from the bipartite disease-symptoms network from[50]. Similarly to[50], we use the cosine distance to compute the symptom-based diseases similarity for this network. We kept for the disease-disease network all interactions with a cosine distance superior to 0.5 iii) A comorbidity network from epidemiological data extracted from[51].
- *Human molecular multiplex network (MOL)* is a molecular network, consisting of 3 layers: (i) A protein-protein interaction (PPI) layer corresponding to the fusion of 3 datasets: APID (apid.dep.usal.es) (Level 2, human only), Hi-Union and Lit-BM (http://www.interactome-atlas.org/download). (ii) A pathways layer extracted from NDEx[52] and corresponding to the human Reactome data[53]. iii) A molecular complexes layer constructed from the fusion of Hu.map[54] and Corum[55], using OmniPathR[56].

Methods implemented for comparisons. We compare MultiVERSE with 6 methods designed for monoplex network embedding (deepwalk, node2vec, LINE) and multiplex network embedding (Ohmnet, MNE, Multi-node2vec), and 4 link prediction heuristic scores (only in the link prediction task).

Monoplex network embedding methods.

- *deepwalk*[5]: This method is based on non-biased random walks, and apply the skip-gram algorithm[20] to learn the embeddings. We set the context window to 10, and the number of random walks to start at each node to 10.
- *node2vec*[6]: This method is an extension of deepwalk with a pair of parameters p and q that biases the random walks for Breadth-first Sampling or Depth-first Sampling. We set $p = 2$ and $q = 1$ to promote moderate explorations of the random walks from a node, as stated in[6]. We set the other parameters as for deepwalk.
- *LINE*[19]: LINE is not based on random walks, but computes the similarities using an adjacency-based proximity measure in association with negative sampling. It approximates the first and second order proximities in the network from one node. First order proximity refers to the local pairwise proximity between the nodes in the network (only neighbours), and second order proximity look for nodes sharing many connections. We set the negative ratio to 5.

Multiplex network embedding methods.

- *OhmNet*[13]: This approach takes into account the multi-layer structure of multiplex networks. It is a random walk-based method that uses node2vec to learn the embeddings layer by layer. We applied the same parameters as in node2vec. The user has to define a hierarchy between layers. We created a 2-level hierarchy for all multiplex networks with first layer as the higher in the hierarchy and the other layers are defined at the second level of the hierarchy, in the same way as[11].
- *MNE*[12]: This method is also designed for multiplex networks and uses node2vec to learn the embeddings layer by layer. For each node, MNE computes a high-dimensional common embedding and a lower-dimensional additional embedding for each type of relation of the multiplex network. The final embedding is computed using a weighted sum of these two high-dimensional and low-dimensional embeddings. We used the default parameters (https://github.com/HKUST-KnowComp/MNE).
- *Multi-node2vec*[14]: This multiplex network embedding method is also based on node2vec. The random walks can jump to different layers and explore in this way the multiplex neighborhood. The length of the random walks is set to 100.

We used OpenNE (https://github.com/thunlp/OpenNE) to implement deepwalk, node2vec and LINE. The other methods have been implemented from the source code associated to the different publications.

Link prediction heuristics. In order to evaluate the relevance of the aforementioned network embedding methods, we also compared them with four classical and straightforward link prediction heuristic scores for node pairs[6]. Table 2 provides formal definitions of these heuristic scores.

Evaluation tasks. On multiplex networks, we evaluate the different methods by measuring their performances in two different tasks: link prediction and network reconstruction. For all the evaluations, we set the embedding dimension to $d = 128$ as in[5,6,13] for fair comparisons, and used the package EvalNE v0.3.1[57]. EvalNE is a package dedicated to the evaluation of network embedding.

From node embeddings to edges. MultiVERSE and the other embedding methods allow learning vector representations of nodes from networks. We aim here to test their performance on link prediction and network reconstruction. We hence need to predict whether an edge exists between every pairs of node embeddings. To do so, given two nodes u and v, we define an operator $\circ$ over the corresponding embeddings $f(u)$ and $f(v)$. This gives $g : V \times V \rightarrow \mathbb{R}^d$, with d the dimension of the embeddings, V the set of nodes and $g(u, v) = f(u) \circ f(v)$. Our test network contains both true and false edges (present and absent edges, respectively). We apply five different operators $\circ$: Hadamard, Average, Weighted-L1, Weighted-L2 and Cosine (Table 3)).

The outputs of the embedding operators are used to feed a binary classifier for the evaluation tasks. This classifier aims to predict if there is an edge or not between two nodes embeddings. Similarly, we use the output of the four link prediction heuristic scores described in Table 2 with a binary classifier to predict edges in a multiplex network.

Link prediction. We first evaluate the performance of the different methods to predict links removed from the original multiplex networks (Fig. 3). We remove 30% of the links in each layer of the original networks. We applied the Andrei Broder algorithm[58] in order to randomly select the links to be removed while keeping a connected graph in each layer. This step provides the multiplex training network, to which we apply the 3 categories of methods (see Fig. 3):

Score	Definition
Jaccard Coefficient (JC)	$\frac{\lvert\mathcal{N}(u)\cap\mathcal{N}(v)\rvert}{\lvert\mathcal{N}(u)\cup\mathcal{N}(v)\rvert}$
Common neighbours (CN)	$\lvert\mathcal{N}(u)\cap\mathcal{N}(v)\rvert$
Adamic Adar (AA)	$\sum_{t\in\lvert\mathcal{N}(u)\cap\mathcal{N}(v)\rvert}\frac{1}{\log\lvert\mathcal{N}(t)\rvert}$
Preferential attachment (PA)	$\lvert\mathcal{N}(u)\rvert.\lvert\mathcal{N}(v)\rvert$

Table 2. Definition of the heuristic scores of a link (u, v) in the graph $G(V, E)$. $\mathcal{N}(u)$ denotes the set of neighbour nodes of node $u \in V$ in $G(V, E)$.

Operators	Symbol	Definition
Hadamard	$\odot$	$[f(u)\odot f(v)]_i = \frac{f_i(u)*f_i(v)}{2}$
Average	$\boxplus$	$[f(u)\boxplus f(v)]_i = f_i(u) + f_i(v)$
Weighted-L1	$\lVert\,.\,\rVert_{\overline{1}}$	$\lVert f_i(u), f_i(v)\rVert_{\overline{1}}\, i = \lvert f_i(u) - f_i(v)\rvert$
Weighted-L2	$\lVert\,.\,\rVert_{\overline{2}}$	$\lVert f_i(u), f_i(v)\rVert_{\overline{2}}\, i = \lvert f_i(u) - f_i(v)\rvert^2$
Cosine	*cos*	$cos[f(u), f(v)]_i = \frac{f_i(u)*f_i(v)}{\lVert f_i(u)\rVert\lVert f_i(v)\rVert}$

Table 3. Embedding operators used to predict edges in the tasks of link prediction and network reconstruction. The definitions describe the i^{th} components of $g(u, v)$.

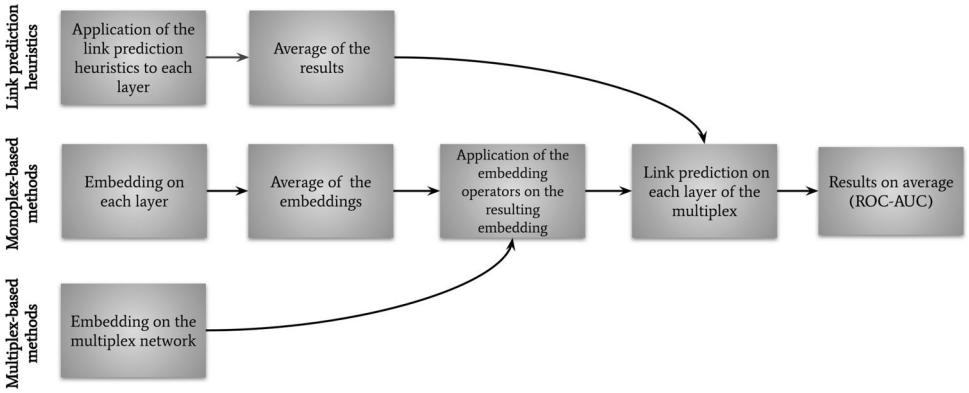

Figure 3. General approach for link prediction on multiplex networks: (top) for the link prediction heuristics, we apply them to each layer and average them across all layers; (center) for monoplex-based methods, we embed each layer with the given method, then average it; (bottom) for multiplex-based methods, we apply the specific embedding method to the network. The embedding operators are then applied to monoplex- and multiplex-based method embeddings. The three types of methods are finally evaluated for link prediction using a binary classifier and a ROC-AUC is computed.

- The methods specifically designed for monoplex network embedding (node2vec, deepwalk and LINE) are applied individually on each layer of the multiplex networks. We thereby obtain one embedding per layer and average them (arithmetic mean) in order to obtain a single embedding for each node. We then apply the embedding operators. We refer to these approaches in the results section as node2vec-av, deepwalk-av and LINE-av.
- Methods specifically designed for multiplex network embedding (Ohmnet, MNE, Multi-node2vec) are applied directly on the training multiplex network. We then apply the embedding operators.
- The link prediction heuristic scores JC, CN, AA and PA are applied individually on each layer of the multiplex networks. We then average the scores, as JC-av, CN-av, AA-av, and PA-av.

From the outputs of the embedding operators and heuristic scores, we feed and train a binary classifier and then test it on the 30% of test edges that have been removed initially. The binary classifier is a logistic regressor.

The evaluation metrics for link prediction is ROC-AUC as it is commonly used for embedding evaluation on link prediction and to validate network embedding[6,12]. The ROC-AUC is computed as the area under the ROC curve, which plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. An AUC value of 1 represent a model that classifies perfectly the samples.

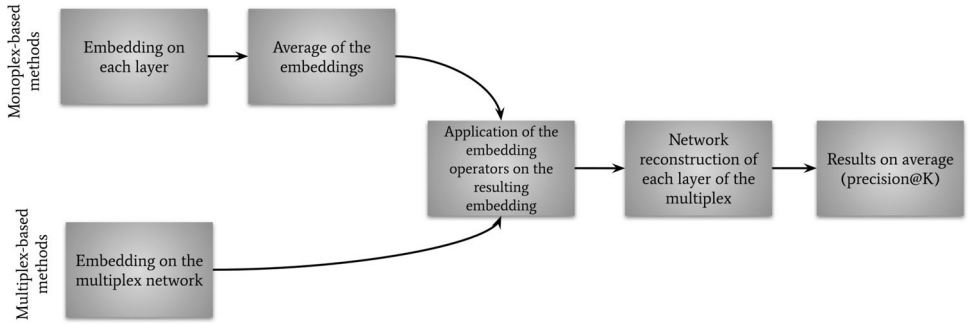

Figure 4. General approach for network reconstruction on multiplex networks: (top) for monoplex-based methods, embed each layer with the given method, then average it; (bottom) for multiplex-based methods, apply the specific embedding method to the network. Embedding operators are then applied to monoplex- and multiplex-based method embeddings. The three types of methods are finally evalutated for network embedding using a binary classifier and a precision@K score is computed.

Network reconstruction. Network reconstruction is another approach to evaluate network embedding methods[11,59,60]. In this case, the goal is to quantify the amount of topological information captured by the embedding methods. This is equivalent to predict if we can go back from the embedding to the original adjacency matrix of each layer of the multiplex network.

Theoretically, to reconstruct the networks, one would need to apply link prediction to every possible edge in the graphs. This is however in practice not scalable to large graphs. Indeed, it would correspond to $n(n-1)/2$ potential edges to classify (for undirected networks of n nodes without self-loops). In addition, the networks in our study are sparse, with much more false (absent) than true (present) edges, leading to large class imbalance. In this context, ROC-AUC can be misleading, as large changes in the ROC Curve or ROC-AUC score can be caused by a small number of correct or incorrect predictions[61]. In order to account for class imbalance, we used the precision@K[59]. This evaluation metric is based on the sorting in descending order of all predictions and consider the first K best predictions to evaluate how many true edges (the minority class) are predicted correctly by the binary classifier. From the outputs of the embedding operators, we perform network reconstruction by training a binary classifier on a subset of the original networks (Fig. 4). We choose a subset of 95% of the edge pairs from the original adjacency matrix of each layer for the smaller multiplex networks (CKM, LAZEGA and C.ELE) to construct the training graph. As the class imbalance increases with the number of nodes and sparsity of the networks, we choose smaller subsets for the largest networks, respectively 5% of edges for the ARXIV network and 2.5% for the other networks, as in previous publications[59,60]. For each layer, K is defined as the maximum of true edges in this subset of edge pairs. We use a Random Forest algorithm as a binary classifier for network reconstruction, as it is known to be less sensitive to class imbalance[62]. In network reconstruction, the results correspond to the training phase of the classifier, there is no test phase.

Evaluation of multiplex-heterogeneous network embedding. *Multiplex-heterogeneous network datasets.*

- *Gene-disease multiplex-heterogeneous network* We use the two multiplex networks presented in the previous sections: the disease (DIS) and molecular multiplex networks (MOL) (Table 1). In addition, we extracted the curated gene-disease bipartite network from the DisGeNET database[63] in order to connect the two multiplex networks. This bipartite interaction network contains 75445 interactions between 5188 diseases and 9179 genes. We obtain a multiplex-heterogeneous network, as represented in Fig. 1C.
- *Drug-target multiplex-heterogeneous network* We use the molecular multiplex network (MOL) from the previous multiplex-heterogeneous network. We constructed the following 3-layers drug multiplex network: (i) the first layer (2795 edges, 877 nodes) has been extracted from Bionetdata (https://rdrr.io/cran/bionetdata/man/DD.chem.data.html) and the edges correspond to Tanimoto chemical similarities between drugs if superior to 0.6, (ii) the second layer (678 edges, 362 nodes) comes from[64] and the edges are based on drug combinations as reported in clinical data, (iii) the third layer (13397 edges, 658 nodes) is the adverse drug-drug interactions network available in[64]. The drug-target bipartite network has been extracted from the same publication[64], and contains 15030 bipartite interactions between 4412 drugs and 2255 protein targets.

Evaluation task. We validate the multiplex-heterogeneous network embedding using link prediction. We remove randomly 30% of the edges but only from the bipartite interactions to obtain a training graph. We then train a Random Forest on the training graph, and test on the 30% removed edges. Based on the multiplex-heterogeneous networks described previously, the idea behind this evaluation is to test if we can predict gene-disease and drug-gene links. Comparisons with other approaches are not possible as, to our knowledge, no existing multiplex-heterogeneous network embedding method are currently available in the literature.

Operators	Method	CKM	LAZEGA	C.ELE	ARXIV	DIS	HOMO	MOL
Link prediction heuristics	CN-av	0.4944	0.6122	0.5548	0.5089	0.5097	0.5113	0.5408
	AA-av	0.4972	0.6105	0.549	0.5081	0.5428	0.5112	0.5404
	JC-av	0.4911	0.523	0.5424	0.5113	0.5425	0.5113	0.5433
	PA-av	0.5474	0.6794	0.5634	0.5139	0.496	0.5185	0.5278
Hadamard	node2vec-av	0.7908	0.6372	0.8552	0.9775	0.9093	0.8638	0.8753
	deepwalk-av	0.7467	0.6301	0.8574	0.9776	0.9107	0.8638	0.8763
	LINE-av	0.5073	0.4986	0.5447	0.8525	0.9013	0.8852	0.8918
	Ohmnet	0.7465	0.7981	0.833	0.9605	0.9333	0.9055	0.8613
	MNE	0.5756	0.6356	0.794	0.9439	0.9099	0.8313	0.8736
	Multi-node2vec	0.8182	0.7884	0.8375	0.9581	0.8528	0.8592	0.8835
	MultiVERSE	0.8177	***0.8269***	***0.8866***	***0.9937***	***0.9401***	***0.917***	***0.9259***
Weighted-L1	node2vec-av	0.7532	0.737	0.8673	0.9738	0.885	0.6984	0.7976
	deepwalk-av	0.7226	0.7094	0.8635	0.9751	0.8888	0.7142	0.8089
	LINE-av	0.6091	0.5776	0.6192	0.7539	0.8586	0.7439	0.7792
	Ohmnet	0.7421	0.7849	0.8128	0.8488	0.8503	0.7007	0.6983
	MNE	0.6289	0.6523	0.8019	0.7805	0.8313	0.7619	0.8182
	Multi-node2vec	0.8611	0.8089	0.8261	0.9659	0.8628	0.8472	0.8997
	MultiVERSE	0.7043	0.7789	0.7516	0.8647	0.7754	0.683	0.7273
Weighted-L2	node2vec-av	0.7556	0.6851	0.8691	0.9743	0.8867	0.7048	0.8028
	deepwalk-av	0.7221	0.6904	0.864	0.9771	0.8891	0.7145	0.813
	LINE-av	0.5851	0.5756	0.6275	0.7609	0.8621	0.7429	0.7835
	Ohmnet	0.7505	0.7788	0.8166	0.8439	0.8599	0.7041	0.6992
	MNE	0.601	0.5397	0.7999	0.7815	0.8333	0.7483	0.8122
	Multi-node2vec	***0.8637***	0.8091	0.8282	0.968	0.8675	0.8525	*0.9004*
	MultiVERSE	0.7125	0.7801	0.7441	0.8661	0.7808	0.6918	0.7475
Average	node2vec-av	0.59	0.6596	0.6842	0.6615	0.8256	0.8308	0.777
	deepwalk-av	0.5954	0.657	0.6784	0.6582	0.8267	0.8307	0.7737
	LINE-av	0.5465	0.6581	0.6699	0.6465	0.8477	0.8653	0.8276
	Ohmnet	0.5764	0.656	0.7334	0.6772	0.8533	0.8825	0.7962
	MNE	0.5882	0.6615	0.7028	0.6723	0.8242	0.8024	0.783
	Multi-node2vec	0.5571	0.6584	0.7365	0.6657	0.8222	0.8216	0.7589
	MultiVERSE	0.5963	0.6728	0.7438	0.6752	0.8586	0.8643	0.812
Cosine	node2vec-av	0.7805	0.7335	0.8515	0.9711	0.8643	0.7368	0.8105
	deepwalk-av	0.7465	0.7066	0.8416	0.9724	0.8667	0.7512	0.8079
	LINE-av	0.545	0.5126	0.5477	0.8198	0.7409	0.6745	0.816
	Ohmnet	0.7898	0.7352	0.8094	0.9642	0.859	0.7829	0.7909
	MNE	0.6203	0.6506	0.7877	0.8951	0.8347	0.6474	0.8102
	Multi-node2vec	*0.8532*	0.7931	0.7815	0.9435	0.7151	0.8477	0.8884
	MultiVERSE	0.8148	0.8171	0.8719	0.9909	0.8775	0.8776	0.9103

Table 4. ROC-AUC scores for link prediction on the 7 reference multiplex networks, for link prediction heuristics (CN-av, AA-av, JC-av, PA-av) and network embedding methods combined with different operators (Hadamard, Weighted-L1, Weighted-L2, Average, and Cosine). For each multiplex network, the best score is in bold; for each operator, the best scores are underlined. Overall, the MultiVERSE algorithm combined with the Hadamard shows the best scores.

Case study: discovery of new gene-disease associations. *Link prediction.* Our aim for this case-study is to predict new gene-disease links. We thereby applied MultiVERSE on the full gene-disease multiplex-heterogeneous network without removing any edges, and trained a binary classifier (Random Forest) using edges from the bipartite interactions. Then, we test all possible gene-disease edges that are not in the original bipartite interactions and involve Progeria and Xeroderma pigmentosum VII disease nodes. Finally, we select the top 5 new gene-disease associations for each disease.

Clustering. We also applied MultiVERSE to the gene-disease multiplex-heterogeneous gene-disease network, followed by spherical K-means[65] to cluster the vector representations of nodes. Spherical K-means clustering is well-adapted to high-dimensional clustering[66]. We define the number of clusters for spherical K-means to 500, in order to obtain cluster sizes that can be analysed from a biological point of view.

Results

Evaluation results for multiplex network embeddings. *Link prediction.* We evaluate the performance of the different methods (link prediction heuristics and network embedding) on the task of link prediction applied to the set of multiplex networks. First, we can observe that the heuristics are not efficient for link prediction, with ROC-AUC only slightly better than random classification (Table 4).

The methods based on embedding always perform better than the heuristic baselines. In addition, the ROC-AUC is in most of the cases higher when the models take into account the multiplex network structure rather than the monoplex-average, as observed in[12]. For instance, using the Hadamard operator, the ROC-AUC average over all the networks of the three monoplex-average approaches (node2vec-av, deepwalk-av, LINE-av) is 0.8025, whereas the average of the three multiplex-based approaches (Ohmnet, MNE, Multi-node2vec) is 0.8381. The ROC-AUC score average of MultiVERSE in this context is 0.9011. Nevertheless, node2vec-av and deepwalk-av perform very well and even outperform multiplex-based approaches on various scenarios, for instance link prediction on the C.ELE and ARXIV networks.

MultiVERSE combined with the Hadamard operator outperforms the other methods for all the tested networks but CKM. In addition, MultiVERSE is the best approach when combined with three out of five operators (Hadamard, Average, Cosine). These results suggest that RWR-M is able to better capture the topological features of the networks under study.

Network reconstruction. We next evaluate the performances of the different embedding methods on the task of network reconstruction applied to multiplex networks We now rely on the evaluation metric, precision@K. The experimental results are shown in Table 5.

On one hand, for the small networks (i.e., CKM, LAZEGA and C.ELE), the best precision is achieved with LINE-av in combination with any of the operators but Cosine. In particular, LINE-av obtains a perfect score for the CKM network using the Weighted-L2 or Hadamard operators. MNE is in second position with more than 99% of precision using the Weighted-L1 or Weighted-L2 operators. LINE-av also presents good performances for the C.ELE network with a precision of 93.67% using the Weighted-L2 operator, almost 20% higher than the second best method on this network (Multi-node2vec with a score of 0.7568 using the Weighted-L2 operator).

On the other hand, we can group together the results obtained for the large networks (DIS, ARXIV, HOMO and MOL). In this case, MultiVERSE achieves the best performance in combination with different operators. Large networks are sparse, leading to high class imbalance. Still, MultiVERSE achieves a good score for the HOMO and DIS networks, with precision@K of 0.8729 and 0.6784, respectively. The precision obtained on the molecular network (MOL) is the lowest, with a precision@K of 0.4143. The complexity of the task is possibly higher as the number of nodes and class imbalance increase.

Overall, the lowest scores are obtained by MNE and, in general, the Cosine operator performs poorly for all methods. The network reconstruction process is a complex task, and the performance depends on the size and density of the different layers composing the multiplex network. Nevertheless, MultiVERSE obtains good results for most of the networks without any processing of the imbalanced data.

Evaluation results for multiplex-heterogeneous network embedding. The task of link prediction on multiplex-heterogeneous networks is applied to MultiVERSE only, as to our knowledge no other methods exist for the embedding of multiple nodes from multiplex-heterogeneous networks. MultiVERSE has a score of ROC-AUC superior to 0.9 with the Hadamard and Average operators (Table 6), meaning that the method can predict with high precision the gene-disease and drug-target links from the corresponding multiplex-heterogeneous networks.

Case study results: discovery of new gene-disease associations. *Discovery of new gene-disease associations with link prediction.* The results of the evaluations on multiplex-heterogeneous network link prediction show that MultiVERSE combined with the Hadamard and Average operators reach ROC-AUC scores superior to 0.9 (Table 6). We here investigate in detail the top 5 new gene-disease associations predicted by MultiVERSE combined with these operators for Hutchinson-Gilford Progeria Syndrome (HGPS) and Xeroderma pigmentosum VII (Table 7) .

Hutchinson-Gilford Progeria Syndrome. Hutchinson-Gilford Progeria Syndrome (HGPS) is a rare premature aging genetic disease characterized by postnatal growth retardation, midface hypoplasia, micrognathia, premature atherosclerosis, coronary artery disease, lipodystrophy, alopecia and generalized osteodysplasia[67]. HGPS is caused by mutations in the *LMNA* genes that cause the production of a toxic form of the Lamin A protein called Progerin.

MultiVERSE top predictions reveal interesting candidate genes (Table 7). In particular, *NOS2* encodes a nitric oxide synthase expressed in liver. It has been associated with longevity[68]. *TNF* is a member of the tumor necrosis factor superfamily, and produces a multifunctional proinflammatory cytokine. *TNF* is also known to be involved in aging[69] and has been previously linked to Progeria[70]. *TERF1* and *TERF2* both encode telomere-binding proteins and *TERF2IP* encodes a protein that is part of a complex involved in telomere length regulation. HGPS patients show increased activation of DNA damage signalling at telomeres associated to reduced telomere length[71]. In addition, it has been reported DNA damage accumulation and *TRF2* degradation in atypical Werner syndrome (adult Progeria) fibroblasts with *LMNA* mutations[72]. *POT1* also produces a telomeric protein that has been linked to the Werner syndrome[73], the maintenance of haematopoeitic stem cell activity during aging[74] and cellular senescence[75]. *IL6* encodes a cytokine involved in inflammation, which have also been linked to aging[76].

Operators	Method	CKM (95%)	LAZEGA (95%)	C.ELE (95%)	ARXIV (5%)	DIS (2,5%)	HOMO (2,5%)	MOL (2.5%)
Hadamard	node2vec-av	0.6764	0.9174	0.4526	0.8207	0.5578	0.7599	0.2989
	deepwalk-av	0.6564	0.9351	0.4416	0.7886	0.5486	0.7636	0.3164
	LINE-av	***1.0***	0.9924	0.8924	0.8204	0.4955	0.5191	0.4006
	Ohmnet	0.7842	0.8334	0.5329	0.9156	0.4811	0.6979	0.2591
	MNE	0.9505	0.9094	0.2728	0.7891	0.4218	0.3641	0.1316
	Multi-node-2vec	0.8352	0.8811	0.6875	0.8605	0.6063	0.7584	0.3123
	MultiVERSE	0.9687	0.9695	0.7436	0.9015	0.6734	***0.8729***	0.3674
Weighted-L1	node2vec-av	0.5923	0.9494	0.5129	0.6922	0.5859	0.8123	0.3194
	deepwalk-av	0.5791	0.9784	0.4896	0.6878	0.5921	0.7984	0.3206
	LINE-av	0.9985	0.9953	0.9229	0.7837	0.4921	0.6839	0.3586
	Ohmnet	0.7355	0.8581	0.5785	0.8771	0.6025	0.8019	0.3769
	MNE	0.9926	0.975	0.4722	0.8593	0.4377	0.5241	0.1861
	Multi-node-2vec	0.8636	0.9235	0.7379	0.7684	0.6356	0.7649	0.2671
	MultiVERSE	0.8545	0.9638	0.7444	0.8705	0.6678	0.7913	0.3559
Weighted-L2	node2vec-av	0.5886	0.9436	0.5097	0.6983	0.5953	0.8193	0.352
	deepwalk-av	0.5829	0.9672	0.5146	0.6877	0.5857	0.805	0.3233
	LINE-av	***1.0***	***0.9962***	***0.9367***	0.7749	0.4945	0.6697	0.392
	Ohmnet	0.7418	0.8687	0.5724	0.8694	0.6209	0.8143	0.3701
	MNE	0.9926	0.9764	0.4646	0.8818	0.4351	0.5529	0.176
	Multi-node-2vec	0.8644	0.93	0.7568	0.7548	0.6361	0.7896	0.2922
	MultiVERSE	0.8653	0.969	0.754	0.8776	***0.6784***	0.7876	0.3701
Average	node2vec-av	0.8408	0.917	0.4817	0.889	0.5587	0.6809	0.2686
	deepwalk-av	0.8331	0.9379	0.501	0.8853	0.5318	0.6714	0.2795
	LINE-av	0.9855	0.9382	0.7103	0.8725	0.5093	0.5677	0.3244
	Ohmnet	0.9412	0.8287	0.5825	0.906	0.4989	0.6551	0.2887
	MNE	0.9179	0.9151	0.2966	0.7146	0.4175	0.352	0.1444
	Multi-node-2vec	0.9767	0.8937	0.6726	0.9498	0.6243	0.6216	0.2901
	MultiVERSE	0.978	0.9059	0.5326	***0.9758***	0.6316	0.7204	***0.4143***
Cosine	node2vec-av	0.5103	0.4936	0.18	0.2537	0.1825	0.116	0.0441
	deepwalk-av	0.4807	0.4776	0.1741	0.2835	0.1854	0.1036	0.0462
	LINE-av	0.3291	0.4974	0.1867	0.2638	*0.2384*	0.1476	0.0454
	Ohmnet	0.5696	0.509	0.1718	0.2655	0.1984	0.1311	0.044
	MNE	0.3169	0.4536	0.1768	0.2445	0.1957	0.1667	0.044
	Multi-node-2vec	0.5127	0.52	0.186	0.273	0.195	0.1032	0.0461
	MultiVERSE	0.6395	0.5026	0.1818	0.254	0.1983	0.1522	0.0474

Table 5. precision@K scores for network reconstruction on the 7 reference multiplex networks, for the network embedding methods combined with different embeddings operators (Hadamard, Weighted-L1, Weighted-L2, Average, and Cosine). For each multiplex network, the best score is in bold; for each operator, the best score is underlined. The percentage of edges used for the reconstruction is indicated in parenthesis under the name of the network. In the case of large networks (DIS, ARXIV, HOMO and MOL) MultiVERSE achieves the best performance in combination with different operators.

Operators	Gene-disease bipartite	Drug-target bipartite
Hadamard	**0.95**	**0.9701**
Weighted-L1	0.7962	0.8057
Weighted-L2	0.7951	0.8055
Average	**0.9603**	**0.9703**
Cosine	0.7765	0.8338

Table 6. ROC-AUC scores for link prediction using MultiVERSE on 2 multiplex-heterogeneous reference networks. Link predictions are computed for the bipartite interactions of the multiplex-heterogeneous networks. The scores higher than 0.9 are highlighted in bold.

HGPS		Xeroderma p. VII	
Average	**Hadamard**	**Average**	**Hadamard**
NOS2	*POT1*	*TNF*	*TNF*
IL6	*TERF1*	*SOD2*	*VCAM1*
TNF	*EEF1A1*	*IL6*	*NUP62*
SOD1	*TERF2*	*TP53*	*ERCC2*
SOD2	*TERF2IP*	*FN1*	*MCC*

Table 7. Top 5 predictions of new gene-disease associations for HGPS and Xeroderma pigmentosum VII by MultiVERSE combined with Average and Hadamard operators.

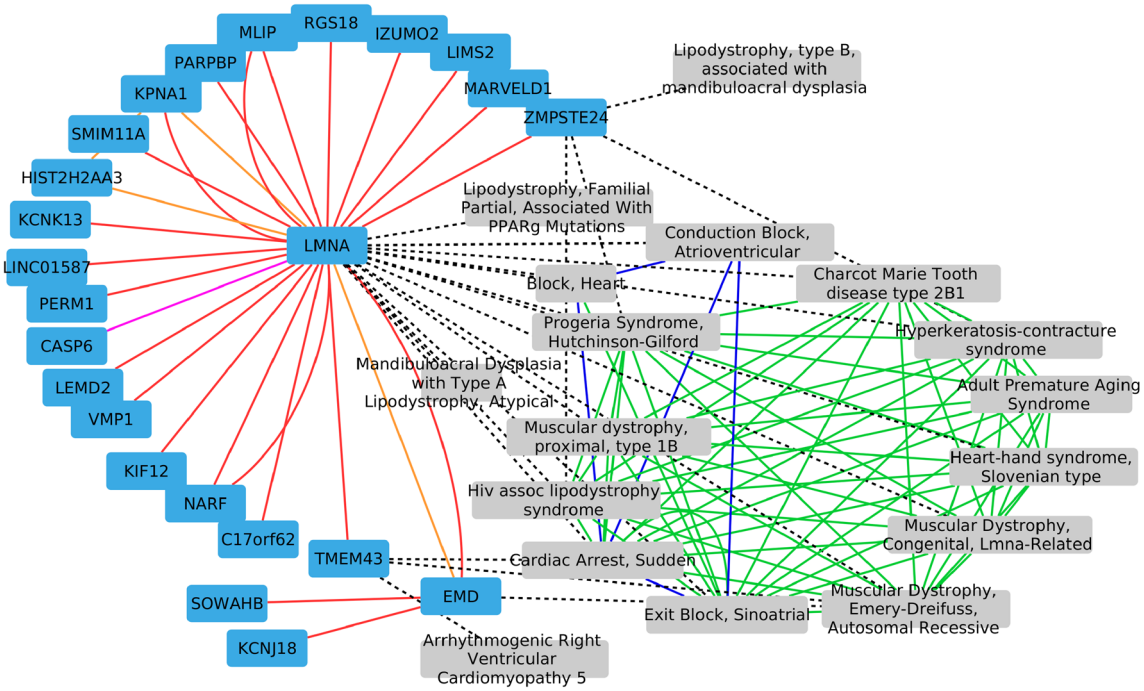

Figure 5. Cluster containing the HGPS disease node. Disease-Disease edges from the disease multiplex network are represented in green (shared symptoms) and blue (CTD projection). Gene-Gene edges from the molecular multiplex network are represented in pink (Reactome pathways), red (protein-protein interactions) and orange (molecular complexes). Gene-Disease bipartite interactions are represented with black dashed lines.

Finally,*SOD1* and *SOD2* are members of the superoxide dismutase multigene family that destroy free superoxide radicals. They both have been associated to aging and cellular senescence[77,78].

Xeroderma pigmentosum VII. Xeroderma Pigmentosum (XP) is characterized by extreme sensitivity to sunlight, resulting in sunburns, pigment changes in the skin and a highly elevated incidence of melanoma. It is a genetically heterogeneous autosomal recessive disorder. Several XP types exist, and the MeSH term Xeroderma pigmentosum VII corresponds to the group G, caused by mutations in the *ERCC5* gene, and with symptoms that overlap Cockayne syndrome[79,80].

MultiVERSE identified various candidates for this disease (see Table 7), we detail here the most interesting ones. *TNF* has been related to XP[81] and skin tumour development[82]. *IL6* is involved in melanoma, one of the major phenotypes of XP[83]. *SOD2*, also predicted as candidate for HGPS, have been recently associated to melanoma[84]. *TP53* is a tumor suppressor implicated in many cancers, in particular melanomas[85]. It has also been associated to XP[86]. *VCAM1* encodes the Vascular Cell Adhesion Molecule, associated to melanoma[87], and *NUP62* encodes the Nuclear pore glycoprotein p62, involved in cell carcinoma proliferation[88]. *ERCC2* produces the *XPD* protein, mutated in XP group D[89].

Discovery of new gene-disease associations with clustering. Another illustration of the advantages of multiplex-heterogeneous network embedding is clustering. We identify clusters with spectral K-means, and focus more particularly on the clusters containing HGPS and Xeroderma pigmentosum VII disease nodes. Clustering is particularly interesting as it can be applied directly on the embeddings without supervised training. In addition,

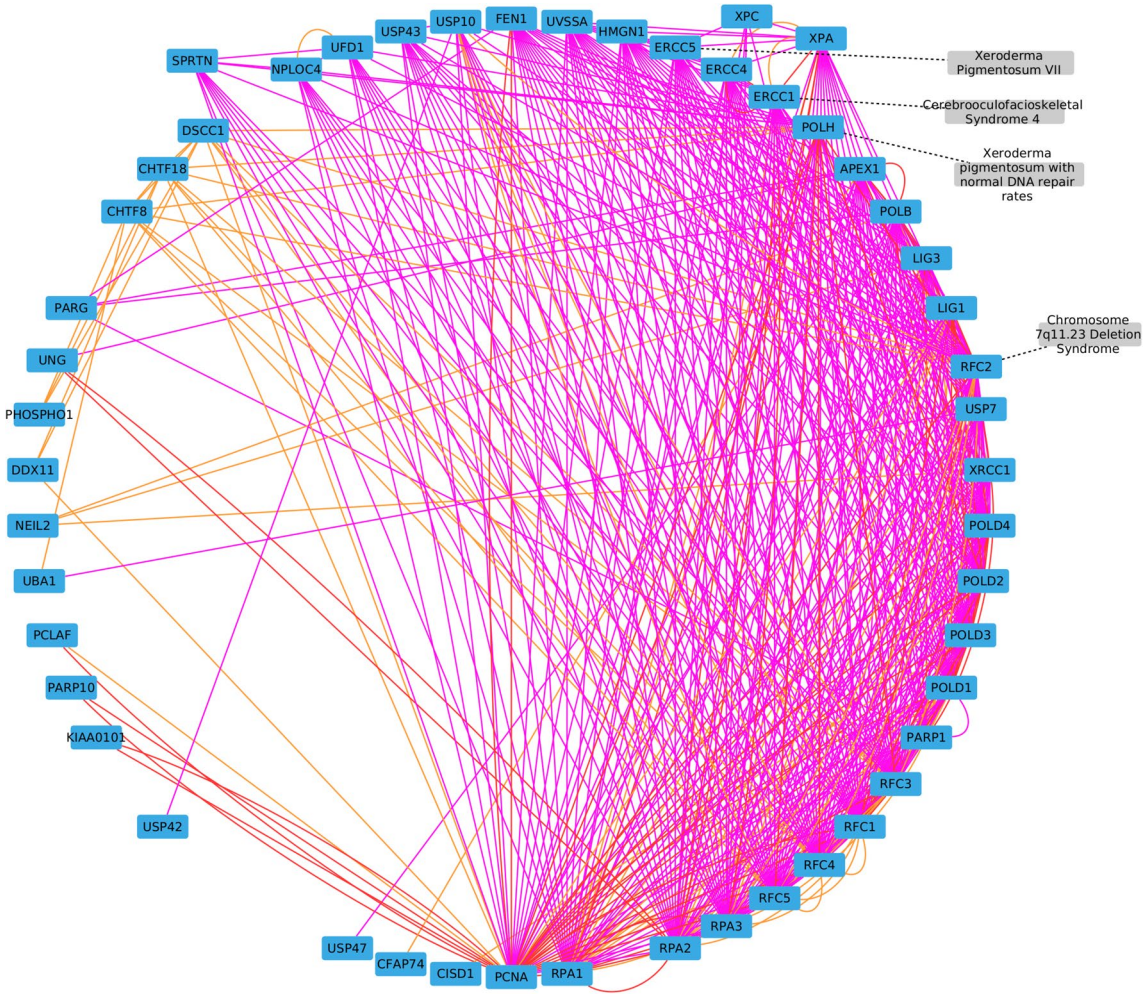

Figure 6. Cluster containing the Xeroderma pigmentosum VII disease node. Gene-Gene edges from the molecular multiplex network are represented in pink (Reactome pathways), red (protein-protein interactions) and orange (molecular complexes). Gene-Disease bipartite interactions are represented with black dashed lines.

it has been shown that clustering from embeddings outperforms the other methods for the detection of biological communities[4].

Cluster containing the HGPS disease node. The cluster containing the HGPS disease node (see Fig. 5) contains the *LMNA* node. *LMNA* mutations have been observed in many diseases that also belong to the identified cluster, including the Heart-hand syndrome (Slovenian type)[90], lipodystrophy associated with mandibuloacral dysplasia[91], the Charcot-Marie-Tooth disease, type 2B1[92], *LMNA*-related muscular distrophy, and different cardiac diseases caused by *LMNA* mutations[93].

We also analysed the cluster's genes annotations with g:Profiler[94] (default parameters). We found significant enrichments in several annotations related to cell nuclear organization. One of the most significant enrichments is nuclear envelope, involving the following genes: *EMD*, *LEMD2*, *LMNA*, *KPNA1*, *MLIP*, *TMEM43*, and *ZMPSTE24*. HGPS is a disorder of the nuclear envelope[95].

Cluster containing the Xeroderma pigmentosum VII disease node. We also analysed the cluster containing the Xeroderma pigmentosum VII disease node (Fig. 6). The cluster contains different diseases, including Xeroderma pigmentosum with normal DNA repair rates, and Cerebro-oculo-facio-skeletal syndrome 4, which is also a nuclear-excision repair disorder[96].

Several genes known for their implication in XP are present in the cluster, such as *ERCC1*, *ERCC4*, *ERCC5*, *XPA* and *XPC*[97]. Using the complete list of genes in the cluster as an input for g:Profiler[94] (default parameters), we identified several significantly enriched annotations. Among them, we can cite nucleotide-excision DNA repair, defective DNA repair after ultraviolet radiation damage or response to ultraviolet radiation. XP patients show important impairments in these biological processes[79]. *SPRTN* is another gene of interest. It encodes a metalloprotease that repairs DNA-protein crosslinks. *SPRTN* does not share interactions with genes known to be mutated in XP, but has been shown to be involved in UV sensibilization and cancer[98].

Discussion and conclusion

We present in this study MultiVERSE, a new approach for multiplex and multiplex-heterogeneous network embedding. MultiVERSE is fully parallelized and scalable, even if the current implementation requires the generation of dense matrices, which can raise memory issues when dealing with very large networks.

For multiplex network embedding, we compared MultiVERSE with state-of-the-art methods using link prediction and network reconstruction. We show that MultiVERSE outperforms various methods specifically developed for multiplex network embedding. Our results suggest that there are several advantages to use RWR for the computation of the similarity between nodes: we experimentally demonstrate that methods based on the SkipGram approach with the truncated random walks (such as node2vec-av, deepwalk-av, ohmnet, multi-node2vec and MNE) are less effective to learn node embedding from multiplex networks than MultiVERSE in several contexts. In particular, multi-node2vec, which is an extension of node2vec (and therefore SkipGram) to multiplex networks, shows reduced performance for our evaluation tasks than MultiVERSE. In addition, node2vec-av is also less effective to learn node embeddings from multiplex networks. SkipGram uses a truncated random walk whereas MultiVERSE uses a (non-truncated) global random walk with RWR to compute the similarity. As RWR-M applies a random walk in pseudo-infinite time, it might allow MultiVERSE to effectively capture node properties and a better representation of the topological structure of the multiplex network.

The methods we found in the literature with available code for multiplex network embedding (i.e., ohmnet, multi-node2vec, MNE) are all based on node2vec. Recently, several graph neural networks approaches have been developed for network embedding, including GCN and GAE/VGAE[27,28]. These methods allow learning high quality embeddings for monoplex networks[99]. However, to the best of our knowledge, they have not yet been extended to multiplex or multiplex-heterogeneous network embedding. It will be interesting to compare MultiVERSE with this class of graph neural networks approaches once they will be developed for multiplex and multiplex-heterogeneous network embedding.

A natural extension of this work would be to consider multiplex networks composed of both directed and undirected layers. In a biological context, this would allow considering metabolic and signalling pathways networks into a multiplex structure without losing the information about the information flow. In addition, for the optimization phase, we set a neighborhood parameter N_{max} that depends on the size of the network. A potential improvement could be to develop an adaptive version of the parameter N_{max} that would depend on node topological properties.

For multiplex-heterogeneous network embedding, MultiVERSE allows the embedding of different types of nodes. We demonstrate its effectiveness for link prediction and illustrate its usefulness for the study of gene-disease associations. We here limited the multiplex-heterogeneous network to two multiplex and one bipartite network. Another natural extension of our work would be to generalize RWR for multiplex-heterogeneous for n multiplex networks and $n(n-1)/2$ bipartite linking them ($n \in \mathbb{N}$). Doing so, one could easily integrate many different types of nodes. The previous discussion about directed networks is in addition also valid for multiplex-heterogeneous network embedding.

By integrating different types of edges for multiplex network embedding or by integrating different types of both edges and nodes for multiplex-heterogeneous network embedding, MultiVERSE could have a wide variety of applications in diverse domains such as network biology and medicine, social science, computer science, neuroscience or physics. Our illustration of MultiVERSE embedding to study gene-disease associations could easily be applied to drug repositioning and drug discovery, for instance with a multiplex drug-drug network, a drug-target bipartite and a molecular multiplex. In this way, genes, diseases and drugs could be projected in the same vector space for further studies. In neuroscience, multiplex-heterogeneous network embedding could be applied to study the links between genes and neurons[100]. In social science, multiplex networks are gaining interest to understand human behaviour[101]. Multiplex-heterogeneous network embedding could give insights on epidemic spread[102,103], socio-economic systems[104] or socio-ecological systems[105].

Received: 13 January 2021; Accepted: 6 April 2021
Published online: 22 April 2021

References

1. Hamilton, W. L. & Ying, R., Leskovec, J. *Methods and applications*. IEEE Data Engineering Bulletin, Representation learning on graphs, (2017).
2. Liao, L., He, X., Zhang, H. & Chua, T.-S. Attributed social network embedding. *IEEE Trans. Knowl. Data Eng.* **30**, 2257–2270 (2018).
3. Ma, G., Lu, C.-T., He, L., Philip, S. Y. & Ragin, A. B. Multi-view graph embedding with hub detection for brain network analysis. In *2017 IEEE International Conference on Data Mining (ICDM)* 967–972 (IEEE, 2017).
4. Nelson, W. *et al.* To embed or not: Network embedding as a paradigm in computational biology. *Front. Genet.* **10**, 381 (2019).
5. Perozzi, B., Al-Rfou, R. & Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* 701–710 (ACM, 2014).
6. Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* 855–864 (ACM, 2016).
7. Kivelä, M. *et al.* Multilayer networks. *J. Complex Netw.* **2**, 203–271. https://doi.org/10.1093/comnet/cnu016 (2014).
8. Valdeolivas, A. *et al.* Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics* **35**, 497–505 (2018).
9. Luo, Y. *et al.* A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun.* **8**, 1–13 (2017).
10. Zhou, D., Orshanskiy, S. A., Zha, H. & Giles, C. L. Co-ranking authors and documents in a heterogeneous network. In *Seventh IEEE international conference on data mining (ICDM 2007)* 739–744 (IEEE, 2007).
11. Bagavathi, A. & Krishnan, S. Multi-net: A scalable multiplex network embedding framework. In *International Conference on Complex Networks and their Applications* 119–131 (Springer, 2018).

12. Zhang, H., Qiu, L., Yi, L. & Song, Y. Scalable multiplex network embedding. *IJCAI* **18**, 3082–3088 (2018).
13. Zitnik, M. & Leskovec, J. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* **33**, i190–i198 (2017).
14. Wilson, J. D., Baybay, M., Sankar, R. & Stillman, P. Fast embedding of multilayer networks: An algorithm and application to group fmri. arXiv preprint arXiv:1809.06437 (2018).
15. Dong, Y., Chawla, N. V. & Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* 135–144 (2017).
16. Shi, C., Hu, B., Zhao, W. X. & Philip, S. Y. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* **31**, 357–370 (2018).
17. Dursun, C., Smith, J. R., Hayman, G. T. & Bozdag, S. Gene Embeddings of Complex network (GECo) and hypertension disease gene classification. *bioRxiv* 10.1101/2020.06.15.149559 (2020). Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2020/06/17/2020.06.15.149559.full.pdf.
18. Tsitsulin, A., Mottin, D., Karras, P. & Müller, E. Verse: Versatile graph embeddings from similarity measures. In *Proceedings of the 2018 World Wide Web Conference*, 539–548 (International World Wide Web Conferences Steering Committee, 2018).
19. Tang, J. *et al.* Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, 1067–1077 (International World Wide Web Conferences Steering Committee, 2015).
20. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
21. Mnih, A. & Hinton, G. E. A scalable hierarchical distributed language model. *Adv. Neural Inf. Process. Syst.* **1081–1088** (2009).
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **3111–3119** (2013).
23. Cao, S., Lu, W. & Xu, Q. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management* 891–900 (2015).
24. Ou, M., Cui, P., Pei, J., Zhang, Z. & Zhu, W. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* 1105–1114 (2016).
25. Liu, X., Murata, T., Kim, K.-S., Kotarasu, C. & Zhuang, C. A general view for network embedding as matrix factorization. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* **375–383** (2019).
26. Hamilton, W. L., Ying, R. & Leskovec, J. Inductive representation learning on large graphs. arXiv preprint arXiv:1706.02216 (2017).
27. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).
28. Kipf, T. N. & Welling, M. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016).
29. Goyal, P. *et al.* Benchmarks for graph embedding evaluation. arXiv preprint arXiv:1908.06543 (2019).
30. Boccaletti, S. *et al.* The structure and dynamics of multilayer networks. *Phys. Rep.* **544**, 1–122 (2014).
31. De Domenico, M., Granell, C., Porter, M. A. & Arenas, A. The physics of spreading processes in multilayer networks. *Nat. Phys.* **12**, 901–906 (2016).
32. Guo, Q., Cozzo, E., Zheng, Z. & Moreno, Y. Levy random walks on multiplex networks. *Sci. Rep.* **6**, 1–11 (2016).
33. Gutmann, M. & Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* **297–304**, (2010).
34. Mnih, A. & Teh, Y. W. A fast and simple algorithm for training neural probabilistic language models. arXiv preprint arXiv:1206.6426 (2012).
35. Lovász, L. Random walks on graphs: A survey. *Combinatorics Paul Erdos is Eighty* **2**, 1–46, 10.1.1.39.2847 (1993).
36. De Domenico, M., Solé-Ribalta, A., Gómez, S. & Arenas, A. Navigability of interconnected networks under random failures. *Proc. Natl. Acad. Sci. USA* **111**, 8351–6. https://doi.org/10.1073/pnas.1318469111 (2014).
37. Battiston, F., Nicosia, V. & Latora, V. Structural measures for multiplex networks. *Phys. Rev. E* **89**, 1–16. https://doi.org/10.1103/PhysRevE.89.032804 (2014).
38. De Domenico, M. *et al.* Mathematical formulation of multilayer networks. *Physical Review X* **3**, 1–15, https://doi.org/10.1103/PhysRevX.3.041022(2013). arXiv:1307.4977v2.
39. Lee, S., Park, S., Kahng, M. & Lee, S. G. PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Syst. Appl.* **40**, 684–697. https://doi.org/10.1016/j.eswa.2012.08.004 (2013).
40. Dursun, C., Shimoyama, N., Shimoyama, M., Schläppi, M. & Bozdag, S. Phenogeneranker: A tool for gene prioritization using complete multiplex heterogeneous networks. *bioRxiv*https://doi.org/10.1101/651000 *(2019). https://www.biorxiv.org/content/early/2019/05/27/651000.full.pdf.*
41. Coleman, J., Katz, E. & Menzel, H. The diffusion of an innovation among physicians. *Sociometry* **20**, 253–270 (1957).
42. Emmanuel, L. The collegial phenomenon. the social mechanisms of cooperation among peers in a corporate law partnership (2001).
43. Chen, B. L., Hall, D. H. & Chklovskii, D. B. Wiring optimization can relate neuronal structure and function. *Proc. Natl. Acad. Sci. USA* **103**, 4723–4728 (2006).
44. De Domenico, M., Porter, M. A. & Arenas, A. Muxviz: A tool for multilayer analysis and visualization of networks. *J. Complex Netw.* **3**, 159–176 (2015).
45. De Domenico, M., Lancichinetti, A., Arenas, A. & Rosvall, M. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X* **5**, 011027 (2015).
46. Stark, C. *et al.* Biogrid: A general repository for interaction datasets. *Nucleic Acids Research* **34**, D535–D539 (2006).
47. Davis, A. P. *et al.* The comparative toxicogenomics database: Update 2019. *Nucleic Acids Res.* **47**, D948–D954 (2019).
48. Zitnik, M., Sosič, R., Maheshwari, S. & Leskovec, J. BioSNAP Datasets: Stanford biomedical network dataset collection. http://snap.stanford.edu/biodata (2018).
49. Hagberg, A., Swart, P. & S Chult, D. Exploring network structure, dynamics, and function using networkx. Tech. Rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008).
50. Zhou, X., Menche, J., Barabási, A.-L. & Sharma, A. Human symptoms-disease network. *Nat. Commun.* **5**, 1–10 (2014).
51. Jensen, A. B. *et al.* Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nat. Commun.* **5**, 1–10 (2014).
52. Pratt, D. *et al.* Ndex, the network data exchange. *Cell Syst.* **1**, 302–305 (2015).
53. Croft, D. *et al.* The reactome pathway knowledgebase. *Nucleic Acids Res.* **42**, D472–D477 (2014).
54. Drew, K. *et al.* Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes. *Mol. Syst. Biol.* **13**, 932 (2017).
55. Giurgiu, M. *et al.* Corum: The comprehensive resource of mammalian protein complexes-2019. *Nucleic Acids Res.* **47**, D559–D563 (2019).
56. Turei, D. *et al.* Integrated intra- and intercellular signaling knowledge for multicelullar omics analysis. *bioRxiv*. https://doi.org/10.1101/2020.08.03.221242 (2020).
57. Mara, A., Lijffijt, J. & De Bie, T. Evalne: A framework for evaluating network embeddings on link prediction. arXiv preprint arXiv:1901.09691 (2019).

58. Broder, A. Z. Generating random spanning trees. In *FOCS*, vol. 89, 442–447 (Citeseer, 1989).
59. Wang, D., Cui, P. & Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* 1225–1234 (2016).
60. Goyal, P. & Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.* **151**, 78–94 (2018).
61. Fernández, A. *et al. Learning From Imbalanced Data Sets* (Springer, Berlin, 2018).
62. More, A. & Rana, D. P. Review of random forest classification techniques to resolve data imbalance. In *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)* 72–78 (IEEE, 2017).
63. Piñero, J. *et al.* The disgenet knowledge platform for disease genomics: 2019 update. *Nucleic Acids Res.* **48**, D845–D855 (2020).
64. Cheng, F., Kovács, I. A. & Barabási, A.-L. Network-based prediction of drug combinations. *Nat. Commun.* **10**, 1–11 (2019).
65. Buchta, C., Kober, M., Feinerer, I. & Hornik, K. Spherical k-means clustering. *J. Stat. Softw.* **50**, 1–22 (2012).
66. Zhong, S. Efficient online spherical k-means clustering. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 5, 3180–3185 (IEEE, 2005).
67. De Sandre-Giovannoli, A. *et al.* Lamin a truncation in Hutchinson–Gilford progeria. *Science* **300**, 2055–2055 (2003).
68. Montesanto, A. *et al.* Common polymorphisms in nitric oxide synthase (nos) genes influence quality of aging and longevity in humans. *Biogerontology* **14**, 177–186 (2013).
69. Davizon-Castillo, P. *et al.* Tnf-α-driven inflammation and mitochondrial dysfunction define the platelet hyperreactivity of aging. *Blood* **134**, 727–740 (2019).
70. Osorio, F. G. *et al.* Nuclear lamina defects cause atm-dependent nf-κb activation and link accelerated aging to a systemic inflammatory response. *Genes Dev.* **26**, 2311–2324 (2012).
71. Decker, M. L., Chavez, E., Vulto, I. & Lansdorp, P. M. Telomere length in Hutchinson–Gilford progeria syndrome. *Mech. Ageing Dev.* **130**, 377–383 (2009).
72. Saha, B. *et al.* Dna damage accumulation and trf2 degradation in atypical werner syndrome fibroblasts with lmna mutations. *Front. Genet.* **4**, 129 (2013).
73. Sowd, G., Lei, M. & Opresko, P. L. Mechanism and substrate specificity of telomeric protein pot1 stimulation of the werner syndrome helicase. *Nucleic Acids Res.* **36**, 4242–4256 (2008).
74. Hosokawa, K. *et al.* The telomere binding protein pot1 maintains haematopoietic stem cell activity with age. *Nat. Commun.* **8**, 1–15 (2017).
75. Li, Y. *et al.* Seryl trna synthetase cooperates with pot1 to regulate telomere length and cellular senescence. *Signal Trans. Target. Ther.* **4**, 1–11 (2019).
76. Maggio, M., Guralnik, J. M., Longo, D. L. & Ferrucci, L. Interleukin-6 in aging and chronic disease: A magnificent pathway. *J. Gerontol. A* **61**, 575–584 (2006).
77. Zhang, Y. *et al.* A new role for oxidative stress in aging: The accelerated aging phenotype in sod1-/- mice is correlated to increased cellular senescence. *Redox Biol.* **11**, 30–37 (2017).
78. Velarde, M. C., Flynn, J. M., Day, N. U., Melov, S. & Campisi, J. Mitochondrial oxidative stress caused by sod2 deficiency promotes cellular senescence and aging phenotypes in the skin. *Aging* **4**, 3 (2012).
79. Kraemer, K. H., Lee, M. M. & Scotto, J. Xeroderma pigmentosum: Cutaneous, ocular, and neurologic abnormalities in 830 published cases. *Arch. Dermatol.* **123**, 241–250 (1987).
80. Vermeulen, W., Jaeken, J., Jaspers, N., Bootsma, D. & Hoeijmakers, J. Xeroderma pigmentosum complementation group g associated with cockayne syndrome. *Am. J. Hum. Genet.* **53**, 185 (1993).
81. Capulas, E. *et al.* Ultraviolet-b-induced apoptosis and cytokine release in xeroderma pigmentosum keratinocytes. *J. Investig. Dermatol.* **115**, 687–693 (2000).
82. Arnott, C. H. *et al.* Expression of both tnf-α receptor subtypes is essential for optimal skin tumour development. *Oncogene* **23**, 1902–1910 (2004).
83. Lu, C., Vickers, M. F. & Kerbel, R. S. Interleukin 6: a fibroblast-derived growth inhibitor of human melanoma cells from early but not advanced stages of tumor progression. *Proc. Natl. Acad. Sci. USA* **89**, 9215–9219 (1992).
84. Yuan, L. *et al.* Braf mutant melanoma adjusts to braf/mek inhibitors via dependence on increased antioxidant sod2 and increased reactive oxygen species levels. *Cancers* **12**, 1661 (2020).
85. Giglia-Mari, G. & Sarasin, A. Tp53 mutations in human skin cancers. *Hum. Mutat.* **21**, 217–228 (2003).
86. Sarasin, A. *et al.* Familial predisposition to tp53/complex karyotype mds and leukemia in dna repair-deficient xeroderma pigmentosum. *Blood* **133**, 2718–2724 (2019).
87. Klemke, M., Weschenfelder, T., Konstandin, M. H. & Samstag, Y. High affinity interaction of integrin $\alpha 4\beta 1$ (vla-4) and vascular cell adhesion molecule 1 (vcam-1) enhances migration of human melanoma cells across activated endothelial cell layers. *J. Cell. Physiol.* **212**, 368–374 (2007).
88. Hazawa, M. *et al.* Rock-dependent phosphorylation of nup 62 regulates p63 nuclear transport and squamous cell carcinoma proliferation. *EMBO Rep.* **19**, 73–88 (2018).
89. Taylor, E. M. *et al.* Xeroderma pigmentosum and trichothiodystrophy are associated with different mutations in the xpd (ercc2) repair/transcription gene. *Proc. Natl. Acad. Sci. USA* **94**, 8658–8663 (1997).
90. Renou, L. *et al.* Heart-hand syndrome of Slovenian type: A new kind of laminopathy. *J. Med. Genet.* **45**, 666–671 (2008).
91. Agarwal, A. K., Kazachkova, I., Ten, S. & Garg, A. Severe mandibuloacral dysplasia-associated lipodystrophy and progeria in a young girl with a novel homozygous arg527cys lmna mutation. *J. Clin. Endocrinol. Metab.* **93**, 4617–4623 (2008).
92. Sinha, J. K., Ghosh, S. & Raghunath, M. Progeria: A rare genetic premature ageing disorder. *Indian J. Med. Res.* **139**, 667 (2014).
93. van Tintelen, J. P. *et al.* High yield of LMNA mutations in patients with dilated cardiomyopathy and/or conduction disease referred to cardiogenetics outpatient clinics. *Am. Heart J.* **154**, 1130–1139 (2007).
94. Raudvere, U. *et al.* g: Profiler: A web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic acids Res.* **47**, W191–W198 (2019).
95. Worman, H. J., Östlund, C. & Wang, Y. Diseases of the nuclear envelope. *Cold Spring Harbor Perspect. Biol.* **2**, a000760 (2010).
96. Graham, J. M. Jr. *et al.* Cerebro-oculo-facio-skeletal syndrome with a nucleotide excision-repair defect and a mutated xpd gene, with prenatal diagnosis in a triplet pregnancy. *Am. J. Hum. Genet.* **69**, 291–300 (2001).
97. Sugasawa, K. Xeroderma pigmentosum genes: functions inside and outside DNA repair. *Carcinogenesis* **29**, 455–465 (2008).
98. Hiom, K. Sprtn is a new player in an old story. *Nat. Genet.* **46**, 1155 (2014).
99. Khosla, M., Setty, V. & Anand, A. A comparative study for unsupervised network representation learning. *IEEE Trans. Knowl. Data Eng.* **33**, 1807–1818 (2019).
100. Badhwar, R. & Bagler, G. Control of neuronal network in caenorhabditis elegans. *PloS ONE* **10**, e139204 (2015).
101. Smith-Aguilar, S. E., Aureli, F., Busia, L., Schaffner, C. & Ramos-Fernández, G. Using multiplex networks to capture the multidimensional nature of social structure. *Primates* **60**, 277–295 (2019).
102. Johnson, C. K. *et al.* Spillover and pandemic properties of zoonotic viruses with high host plasticity. *Sci. Rep.* **5**, 14830 (2015).
103. Liu, Q.-H. *et al.* Measurability of the epidemic reproduction number in data-driven contact networks. *Proc. Natl. Acad. Sci. USA* **115**, 12680–12685 (2018).
104. Saracco, F., Di Clemente, R., Gabrielli, A. & Squartini, T. Detecting early signs of the 2007–2008 crisis in the world trade. *Sci. Rep.* **6**, 1–11 (2016).

105. Lenormand, M. *et al.* Multiscale socio-ecological networks in the age of information. *PloS ONE* **13**, e0206672 (2018).

Acknowledgements
We thank Pr. Alfonso Valencia and the Computational Biology Life Sciences Group at Barcelona Supercomputing Center for the warm welcome and discussions and the use of the MareNostrum supercomputer.

Author contributions
Conceptualization: A.B., L.P-L, A.V.; Methodology: L.P-L., A.V.; Software: L.P-L., L.T., A.V.; Formal analysis and investigation: L.P-L.; Writing-Original draft: L.P-L.; Review and editing: A.B., L.P-L., E.R., L.T., A.V.; Visualization: A.B., L.P-L., L.T.; Supervision: A.B.; Project administration: A.B., E.R.; Funding acquisition: A.B., L.P-L., E.R.

Funding
L.P-L is the recipient of a Short Term Collaboration Grant for HPC 2019 from the Eurolab4HPC consortium. This project has received funding from the Excellence Initiative of Aix-Marseille University- A*Midex, a French 'Investissements d'Avenir' program.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to L.P.-L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.