

# Chapter 10

## Labeled Graphs in Life Sciences—Two Important Applications



Piotr Formanowicz, Marta Kasprzak, and Piotr Wawrzyniak

**Abstract** Life sciences and mathematics are usually considered as quite distant areas of research. But in fact there are close relationships between them, especially in recent years, when computational biology and bioinformatics rapidly evolve. The spectacular developments in the area of biological sciences, particularly those related to sequencing genomes, made evident that an application of formal mathematical and computer science methods is necessary for further discovering the nature of the living world. Among many areas of mathematics being useful in this context, graph theory plays especially important role. It is also worth to remember that, despite the fact that graphs are intensively applied in biology during last three decades, they were used in chemistry (being a basement of molecular biology) more than a century ago. In this chapter a short review of selected applications of labeled graphs in biology and chemistry is given. Some graph theory problems concerning molecules of chemical compounds and DNA sequencing are presented.

**Keywords** Labeled graphs · Molecular graphs · DNA sequencing · Structural formulas

---

P. Formanowicz (✉) · M. Kasprzak · P. Wawrzyniak  
Poznan University of Technology, Institute of Computing Science,  
Piotrowo 2, 60-965 Poznań, Poland  
e-mail: [Piotr.Formanowicz@cs.put.poznan.pl](mailto:Piotr.Formanowicz@cs.put.poznan.pl)

M. Kasprzak  
e-mail: [Marta.Kasprzak@cs.put.poznan.pl](mailto:Marta.Kasprzak@cs.put.poznan.pl)

P. Wawrzyniak  
e-mail: [Piotr.Wawrzyniak@cs.put.poznan.pl](mailto:Piotr.Wawrzyniak@cs.put.poznan.pl)

P. Formanowicz  
Polish Academy of Sciences, Institute of Bioorganic Chemistry,  
Noskowskiego 12/14, 61-704 Poznan, Poland

© Springer Nature Switzerland AG 2022  
S. Zawiślak and J. Rysiński (eds.), *Graph-Based Modelling in Science, Technology and Art*, Mechanisms and Machine Science 107,  
[https://doi.org/10.1007/978-3-030-76787-7\\_10](https://doi.org/10.1007/978-3-030-76787-7_10)

## 10.1 Introduction

Computational biology is an interdisciplinary branch of science which evolves at the intersection of computing science, mathematics and biological sciences, especially molecular biology. Its main goal is to develop mathematical models of biological phenomena and algorithms solving various problems arising when biological processes and objects are analyzed using strict methods. Many of the studied phenomena have discrete nature, hence methods based on various branches of discrete mathematics are often very useful to analyze them. Here, especially important is graph theory, since many of the analyzed biological processes and objects have structure which can be described in a natural way using graphs.

But an application of graph theory in life sciences can be dated a century before an emergence of computational biology. Indeed, in the second half of nineteenth century, graphs were used to describe and analyze molecules of chemical compounds. It was a significant impulse to develop important ideas not only in chemistry but also in graph theory.

In this paper a short review of selected applications of labeled graphs to two groups of life sciences problems is presented. The first group concerns determining structural formulas of chemical compounds. The problem has a quite long history, but it has become more important recently when mass spectrometers are becoming more and more precise and more available in biological laboratories.

The second group of considered problems concerns DNA sequencing by hybridization. It is the problem which brought an attention of computer scientists into molecular biology in the late 1980s and in fact caused a rapid development of computational biology. Many interesting theoretical results have been obtained since that time. Some of them are mentioned in this chapter.

## 10.2 Molecular Graphs—Labeled Graphs in Chemistry

Problems considered with graphs have a long history, dating back to 1736 when Leonhard Euler considered the issue of the Königsberg bridges. The word “graph” appeared later when in 1878, in an article published in *Nature*, James Joseph Sylvester introduced the term “chemigraph” along with its abbreviated version of “graph.” It can be said that the application of graph theory to solving chemistry problems gave birth to this science under the name we know. This close relationship between graphs and chemistry will be presented here.

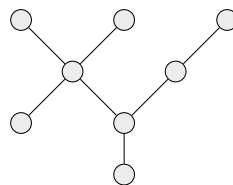
### 10.2.1 Correspondence Between Graph Theory and Chemistry

The possibility of using graphs for the mathematical description of chemical molecules seems obvious when we compare the visual representation of a graph (Fig. 10.1) and the structural formula (Fig. 10.2) of a chemical compound.

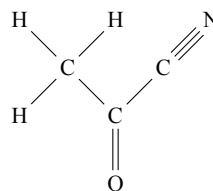
The main differences between these Figs. 10.1 and 10.2 are symbols of elements in a chemical compound. The need to assign different symbols to the vertex brings us to the essential requirement of using a graph to represent a chemical compound. We have to assign a label to each vertex in the graph. After such an operation, each vertex identifies the atom of a specific chemical element in the molecule. Modeling a chemical compound by a graph requires more such relationships between the graph and the chemical compound. There exists more such one-to-one correspondences, like edge–bond, degree of vertex–valence of the atom, and others. They are listed in Table 10.1.

These analogies in constructing a graph and a chemical compound do not exhaust the analogy between graph theory and chemistry. Many problems from the graph theory correspond to the problems of chemistry (see Table 10.2).

**Fig. 10.1** Graph



**Fig. 10.2** Molecule



**Table 10.1** Graph and chemical terms

Graph theory term	Chemical term
Vertex	Atom
Vertex label	Chemical element symbol
Vertex degree	Atom valency
Edge	Chemical bond
Parallel edges/weighted edges	Multiple chemical bonds

**Table 10.2** Graph and chemical problems

Graph theory	Chemistry
Graph isomorphism	Structural isomers
Topological indices (quantitative structure properties)	Chemical or physical properties of substances (e.g. boiling point)
Enumeration problems	Counting all possible molecules for given molecular formula
Planarity	Chirality
Subgraph	A molecule fragment or chemical group

### 10.2.2 Definitions

There are two basic ways to define a molecular graph:

- (i) a labeled multigraph [20]
- (ii) a labeled graph with labeled edges [32].

Each graph can be denoted as  $G = (V, E)$ , where  $V$  is a set of vertices matching the atoms of the compound. Each vertex  $u \in V$  has assigned a label  $l(u)$  matching the chemical element of the atom. Depending on the way of representation,  $E$  is (i) a multiset or (ii) a set of edges. Edge  $e \in E$  is an unordered pair  $\{u, v\} : u, v \in V \wedge u \neq v$ , the inequality of vertex ensures that the graph is a loop free. In the case where the graph is not a multigraph, i.e. (ii), each edge has assigned a label  $w(e)$  which corresponds to the type of a chemical bond.

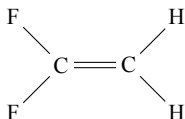
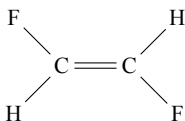
### 10.2.3 Graph Isomorphism and Chemical Isomerism

In chemistry, molecules with the same molecular formulae can be completely different compounds. Because apart from the elemental composition, the bonds between atoms are an essential feature of a chemical compound. Such molecules, made from the same number of atoms of each element but connected differently, are called structural isomers. To investigate whether two molecules are the same, we need to answer the question: *Is the structure of connections between atoms the same?* The answer can be found by checking whether the corresponding molecular graphs are isomorphic or not.

Isomorphism is one-to-one correspondence (a bijection) between vertex sets of two compared graphs,  $G = (V, E)$  and  $G' = (V', E')$ :

$$f : V \rightarrow V' \text{ such that: } \forall_{u,v \in V} \{u, v\} \in E \iff \{f(u), f(v)\} \in E'$$

This definition of isomorphism for the basic graph does not distinguish all differences between molecular graphs. The graphs can have different labels in the

**Fig. 10.3** 1,1-Difluoroethylene**Fig. 10.4** 1,2-Difluoroethylene

same place in the structure. Figure 10.3 (1,1-Difluoroethylene) and Fig. 10.4 (1,2-Difluoroethylene) present two different chemical compounds with the same molecular graph structure.

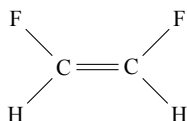
It is easy to notice that these molecular graphs have the same structure but differs in vertex labels. To distinguish between these cases, an extra condition to check label equality must be added to the isomorphism bijection function [25]:

$$f : V \rightarrow V' \text{ such that:}$$

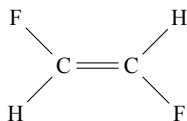
$$(\forall_{u,v \in V} \{u, v\} \in E \iff \{f(u), f(v)\} \in E') \wedge (\forall_{u \in V} l(u) = l(f(u)))$$

Graph isomorphism is a widely studied problem, and many ways to solve it have been proposed. However, no polynomial-time algorithm for this problems has been found. Hence it is not known whether the problem belongs to class P. On the other hand, it was also not possible to confirm its belonging to the class of NP-complete problems [44]. Fortunately, molecular graphs correspond to the natural chemical compounds. They have many limitations, a finite number of labels (corresponding to the symbols of elements), or a limited degree of vertices (corresponding to the valence of an atom in the compound) [1]. Using this knowledge, we can use an isomorphism algorithm for bounded degree graphs, and this problem can be solved in polynomial time [29]. This approach and most other considered graph problems concern simple graphs (without parallel edges and labeled vertices), called them just a “graph”. However, a fast method having polynomial complexity of transforming molecular graphs into simple graphs has been presented [20]. It consists of two steps:

1. Removing parallel edges: each parallel edge is split into two parts by adding an extra vertex between the adjacency source vertices. Such dummy vertices always have a degree equal to 2.
2. Removing labels: each label of a vertex is replaced by attaching a different number of dummy vertices to the labeled vertex. Of course, the number of vertices added this way is unique for each unique label. In this case, the dummy vertices always have a degree equal to 1. For example, the number of added dummy vertices can



**Fig. 10.5** cis-1,2-Difluoroethylene



**Fig. 10.6** trans-1,2-Difluoroethylene

be set to  $Z = Z(a) + Z_0$ , where  $Z(a)$  is the atomic number of  $a$ , and  $Z_0$  is the maximum valence in the compound.

After such transformation, the molecular graph becomes a simple one and still has a limited degree. In such a case, the isomorphism can be checked in polynomial time.

Chemical compounds can have not only the structural isomers, but there are also stereoisomers. This situation occurs for the mentioned earlier 1,2-Difluoroethylene, the substance that has the cis- (Fig. 10.5) and trans- (Fig. 10.6) isomers.

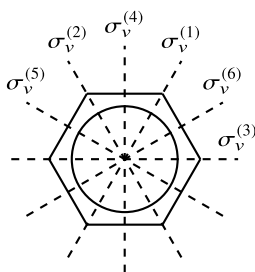
The molecular graph cannot differentiate among such spatial isomers [21]. However, this information (chiral centers, cis/trans isomerism) can be encoded by bonds labels (as in wedge and dash notation) [2]. For part of such problems, e.g., for the cis and trans isomers, a solution has been proposed through the concept of virtual bonds forming virtual cis rings [37].

### 10.2.4 Counting of Molecular Graphs / Isomer Enumeration

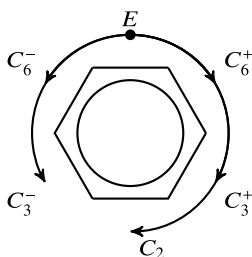
As mentioned earlier, the word graph was introduced in 1878 to denote what is now known as a molecular graph. Even before that date, in 1874, Arthur Cayley first applied the graph theory in chemistry to enumerate alkenes isomers [6]. Moreover, more recently, the DENDRAL [15] enumerate molecules program is called the first expert system [40].

The chemical problem of isomers enumeration is to count the number of different compounds with the same molecular formula. This problem can be solved using graph theory and counting all topologically distinct nodes labelings in the graph [14]. Pólya's enumeration formula gives the most general method of such a calculation.

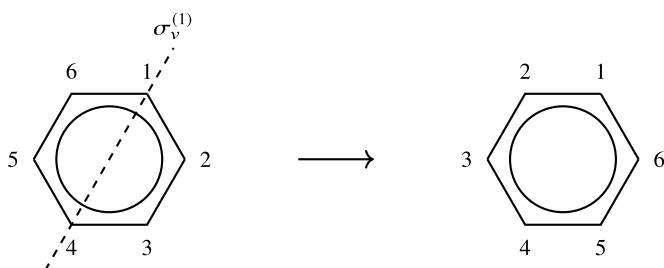
The Pólya algorithm is based on the symmetry recognition of the molecule under study. The method is most often explained on the example of benzene and the substitution of its hydrogens with any other monovalent element, e.g., by fluorine. The



**Fig. 10.7** Symmetry axes



**Fig. 10.8** Symmetry rotations



**Fig. 10.9** Example of  $\sigma_v^{(1)}$  symmetry, permutation group (1)(4)(2 6)(3 5)

six axes of benzene symmetry are shown in Fig. 10.7, while Fig. 10.8 presents six rotations in it, each with a step of  $60^\circ$ . The last 360-degree rotation is identity [35].

Each of these symmetries can be written as a permutation. After each binding site in benzene has been indexed, the applied symmetric operation can be described as a sequence of indexes changes that make up the symmetry's resulting indexing. When vertex with index 1 is not changed, we can denote it as a permutation  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ . When the vertex 2 is replaced with 6 and 6 with 2, it can be marked as a permutation  $\begin{pmatrix} 2 & 6 \\ 6 & 2 \end{pmatrix}$ . Permutations written in two-line notation can be changed to cyclic notation (1) and (2 6). An example permutation group is presented in Fig. 10.9.

**Table 10.3** All symmetries of benzene

Symmetry	Permutation	Cycle index
$E$	(1)(2)(3)(4)(5)(6)	$z_1^6$
$C_6^+$	(1 2 3 4 5 6)	$z_6^1$
$C_6^-$	(6 5 4 3 2 1)	$z_6^1$
$C_3^+$	(1 3 5)(2 4 6)	$z_3^2$
$C_3^-$	(5 3 1)(642)	$z_3^2$
$C_2$	(1 4)(2 5)(3 6)	$z_2^3$
$\sigma_v^{(1)}$	(1)(4)(2 6)(3 5)	$z_1^2 z_2^2$
$\sigma_v^{(2)}$	(3)(6)(1 5)(2 4)	$z_1^2 z_2^2$
$\sigma_v^{(3)}$	(2)(5)(1 3)(4 6)	$z_1^2 z_2^2$
$\sigma_v^{(4)}$	(1 6)(2 5)(3 4)	$z_2^3$
$\sigma_v^{(5)}$	(1 4)(2 3)(5 6)	$z_2^3$
$\sigma_v^{(6)}$	(1 2)(3 6)(4 5)	$z_2^3$

From such notations, we can extract the cycle index needed by Pólya's algorithm. For each permutation  $\alpha$ , shown in Table 10.3, the cycle index  $z_k^i$  describes the number  $i$  of the permutation cycles of a given length  $k$ .

The Pólya's formula  $Z(A)$  is a multivariate polynomial, counting the cycle index for a group of permutations  $A$  [38],

$$Z(A) = \frac{1}{|A|} \sum_{\alpha \in A} \prod_{k=1}^n z_k^{c_k(\alpha)}$$

where:





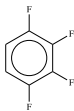
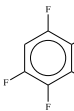
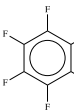

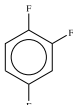

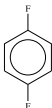

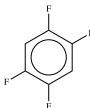
- (i)  $|A|$  is the number of elements in the permutation group  $A$ ,
- (ii)  $\alpha$  is a permutation from group  $A$ ,
- (iii)  $k$  describe the permutation length,
- (iv)  $z_k$  is a variable representing cycles in permutations of the length  $k$ ,
- (v)  $c_k(\alpha)$  is the number of cycles of length  $k$  in permutation  $\alpha$ .

In the product  $\prod_{k=1}^n z_k^{c_k(\alpha)}$ , for each permutation  $\alpha$  the sum of products of cycles length  $k$  and cycles number of such length  $c_k(\alpha)$ , of course, always equals the number of elements in a permutation  $\alpha$ , e.g., for  $z_1^6$   $1 \cdot 6 = 6$ , for  $z_2^3$   $2 \cdot 3 = 6$  or for  $z_1^2 z_2^2$   $1 \cdot 2 + 2 \cdot 2 = 6$ .

Following the Pólya theorem, to obtain number of isomers, every occurrence of variable  $z_k$  has to be replaced by the generating function  $c_k(x)$ . Having only two different chemical elements to choose from (hydrogen and fluorine), the simple generating function  $c_k(x) = 1 + x^k$  can be used [18].



**Table 10.4** All k-fluoro-benzene isomers matched to Pólya's polynomial terms

$1 \times x^0$	$1 \times x^1$	$3 \times x^2$	$3 \times x^3$	$3 \times x^4$	$1 \times x^5$	$1 \times x^6$
						
						
						

Replace in Pólya's formula  $Z(A)$  for benzene example:

$$Z(A) = \frac{1}{12}(z_1^6 + 4z_2^3 + 2z_3^2 + 2z_6^1 + 3z_1^2 z_2^2)$$

each variable  $z_k$  by generating function  $c_k(x) = 1 + x^k$ , results in the following formula:

$$Z(A) = \frac{1}{12}((1+x)^6 + 4(1+x^2)^3 + 2(1+x^3)^2 + 2(1+x^6)^1 + 3(1+x)^2(1+x^2)^2)$$

$$Z(A) = 1 + x + 3x^2 + 3x^3 + 3x^4 + x^5 + x^6$$

The coefficient of the  $x^k$  term in the polynomial describe the number of isomers with  $k$  fluorine atoms (Table 10.4).

### 10.2.5 Molecular Graphs Generation / Isomer Construction

The graphs shown in Table 10.4 are only an illustration presenting the correctness of the Pólya algorithm operation. The construction of such graphs is, however, a task that has many practical applications. One of them is identifying chemical molecules based on experimental data, most often from mass spectrometry. In the simplest case, the mass spectrometry can provide very accurate information about a molecule's mass. This information can then be converted into a simple molecular formula [13] like

$C_6H_4F_2$ . Creating all the structural isomers of such a molecule can be defined as the following graph problem.

The molecular formula is changed to a multiset of labels  $L$  (corresponding to the symbols of elements), and  $d(l)$  function assigning to each label  $l \in L$  a numeric value (corresponding to the valence of the element). The task is to construct a set  $S$  of connected graphs with labels from set  $L$  and corresponding degrees  $d(l)$  that  $\forall_{g,h \in G}$   $g$  and  $h$  are not isomorphic. There are two approaches to solve this problem:

- (i) We create an unlabeled graph basing on the list of given vertices degrees  $d(l)$ . It is a well-known problem of graph realization [23]. Then, to obtain molecular graphs, we assign labels to all unique graphs created in this way. The assignment of these labels also has to keep the graphs topologically distinct. The algorithm for labeling these graphs can, like in the Polya algorithm, use symmetry groups and algebraic structures based on them [14].
- (ii) We start from the empty graph on labeled vertices from a multiset of labels  $L$ . Then, we assign another kind of label to each vertex in the form of an index from 1 to  $|L|$ . In particular the vertex set for the Difluoroethylene ( $C_2F_2H_2$ ) can be presented as  $\{C_1, C_2, F_3, F_4, H_5, H_6\}$ . Next, starting from this empty graph, we extend it by one edge at each step, following the orderly generation algorithm by Read and Faradzev [17, 39]. By introducing the order of added edges, this algorithm eliminates the need to check the created graphs' isomorphism with each other.

Each of these approaches allows the application of additional conditions, e.g., the existence of a specific subgraph, which corresponds to the existence of specific chemical groups in the molecule. These algorithms are applied in various tools [3, 19, 22, 34, 43] used by chemists to discover the chemical structure of unknown compounds, or to prepare libraries of new compounds for biochemical screening.

### 10.3 Sequencing Graphs—Labeled Graphs in Biology

De Bruijn graphs are labeled graphs known for their technical application in modeling communication networks or parallel computer architectures. Their special form well fits such real-world schemas and guarantees a short path between any pair of nodes in a network. What is more, a polynomial-time solution to the problems of the directed Hamiltonian cycle or path, being generally NP-hard, is possible for such graphs. Here, we describe another, biological application of de Bruijn graphs and related classes of digraphs.

### 10.3.1 Definitions

The following statements refer to directed graphs. Graph  $G = (V, A)$  has  $V$  as the set of its vertices and  $A$  as its set of arcs. Arc  $a \in A$  is an ordered pair  $(u, v)$ :  $u, v \in V$ , where  $u$  and  $v$  can be the same vertex. Let  $e(v)$  be the label of vertex  $v$ ,  $\text{suf}_i(s)$  the suffix of length  $i$  of string  $s$ , and  $\text{pre}_i(s)$  the prefix of length  $i$  of  $s$ .

A path in a directed graph is a sequence of vertices  $(v_1, v_2, \dots, v_p)$ , where  $(v_i, v_{i+1}) \in A, i = 1, \dots, p - 1$ . A cycle is a path, for which the condition  $(v_p, v_1) \in A$  is satisfied. A path (cycle) containing every vertex of a graph exactly once is a Hamiltonian path (Hamiltonian cycle). A path (cycle) traversing every arc of a graph exactly once is an Eulerian path (Eulerian cycle).

For an alphabet of size  $\alpha$  and labels of constant length  $k$  ( $k > 1, \alpha > 0$ ), *de Bruijn graph*  $B(\alpha, k) = (V, A)$  has  $\alpha^k$  vertices, every one labeled by a different word over the alphabet. For all  $u, v \in V, (u \neq v \Rightarrow e(u) \neq e(v))$  and  $((u, v) \in A \Leftrightarrow \text{suf}_{k-1}(e(u)) = \text{pre}_{k-1}(e(v)))$  [16]. *Adjoint*  $G = (V, A)$  of a graph  $H = (U, V)$  has vertices corresponding to arcs of  $H$ , and  $(u, v) \in A$  if and only if the head of arc  $u$  in  $H$  is the tail of arc  $v$  [5]. Adjoint  $G$  is always a 1-graph (i.e., it has no multiple arcs),  $H$  need not be. If  $H$  is a 1-graph, its adjoint  $G$  is a *directed line graph* [8].

1-graph  $G = (V, A)$  is an adjoint if and only if, for all  $u, v \in V$ , the following property is satisfied:

$$N^+(u) \cap N^+(v) \neq \emptyset \Rightarrow N^+(u) = N^+(v),$$

where  $N^+(u)$  is the set of immediate successors of vertex  $u$  [5]. 1-graph  $G = (V, A)$  is a directed line graph if and only if the following property is satisfied for all  $u, v \in V$ :

$$N^+(u) \cap N^+(v) \neq \emptyset \Rightarrow N^+(u) = N^+(v) \wedge N^-(u) \cap N^-(v) = \emptyset,$$

where  $N^-(u)$  is the set of immediate predecessors of vertex  $u$  [8]. Digraph  $G = (V, A)$  is a *quasi-adjoint graph* if and only if, for all  $u, v \in V$ , the following property holds [12]:

$$N^+(u) \cap N^+(v) \neq \emptyset \Rightarrow N^+(u) = N^+(v) \vee N^+(u) \subset N^+(v) \vee N^+(v) \subset N^+(u).$$

Quasi-adjoint graphs, unlike adjoints, can be multigraphs.

According to definitions in [8], a directed 1-graph  $G = (V, A)$  belongs to class  $\mathcal{L}_k^\alpha$  (can be  $(\alpha, k)$ -labeled) if it is possible to assign labels to vertices such that, for all  $u, v \in V, (u \neq v \Rightarrow e(u) \neq e(v))$  and  $((u, v) \in A \Leftrightarrow \text{suf}_{k-1}(e(u)) = \text{pre}_{k-1}(e(v)))$ , where  $k > 1$  is the length of labels and  $\alpha > 0$  is the alphabet size. *Labeled graphs* (*uniquely labeled graphs*) are these graphs that belong to a class  $\mathcal{L}_k^\alpha$  for some  $\alpha$  and  $k$ . Graphs satisfying the above requirements except the condition that labels must be different are called *non-uniquely labeled graphs*. A *self-adjoint* is defined as a graph isomorphic to its adjoint [24]. *Alphabet overlap digraphs* are a generalization of de Bruijn graphs. Given three integers,  $\alpha \geq 1, k \geq 2$  and  $1 \leq i < k$ , alphabet overlap digraph  $O(\alpha, k; i) = (V, A)$  is defined as a graph labeled with all possible words of

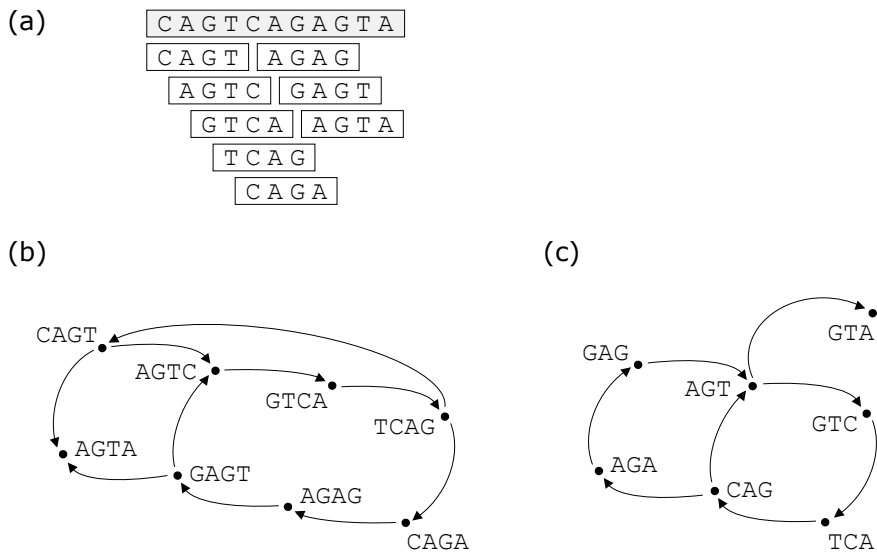
length  $k$  over an alphabet of size  $\alpha$ , where  $i$  is a fixed offset in overlaps of vertex labels [28] (see also [27]). For all  $u, v \in V$ , where  $|V| = \alpha^k$ , ( $u \neq v \Rightarrow e(u) \neq e(v)$ ) and  $((u, v) \in A \Leftrightarrow \text{suf}_{k-i}(e(u)) = \text{pre}_{k-i}(e(v)))$ .

### 10.3.2 Combinatorial Modeling of DNA Sequencing

We refer to one of fundamental issues associated with molecular biology, DNA sequencing. It is a process of recognizing a sequence of nucleotides of a DNA fragment. Such a sequence determines some aspects of functioning of an organism, and knowing it is a first step toward understanding biological mechanisms. Over the years, several techniques were developed to carry through this process, from small-scale laboratory methods [31, 41] to high-throughput automated modern sequencing (e.g., the Illumina sequencing), here we focus on an approach involving an algorithmic stage, the DNA sequencing by hybridization [4, 33, 42].

Not going into biological details (for those see the above references or the algorithmically oriented review [11]), the DNA sequencing by hybridization provides a set  $S$  of short words over the alphabet  $\{‘A’, ‘C’, ‘G’, ‘T’\}$ , where the letters stand for four nucleotides encoding genetic information of an organism: adenine, cytosine, guanine, and thymine. The words are identified via a biological hybridization experiment as parts of a DNA chain, and the goal of the computational problem is to reconstruct the chain from these words. In the case of the classical approach to the hybridization experiment, the words have the same length  $k$  and we call them  $k$ -mers (where  $k$  usually takes values from 8 to 12), they are also assumed to be different within a set. During the algorithmic stage of the process, the words from  $S$  are ordered to obtain a final nucleotide sequence of the examined fragment of a DNA chain (usually of the length a few hundreds of nucleotides). If the hybridization experiment was performed without any error (the theoretical case considered here),  $S$  is complete and the properly ordered words overlap exactly on  $k - 1$  letters in pairs of neighbors, thus they form a sequence of  $|S| + k - 1$  letters.

The computational problem of DNA sequencing by hybridization without any error in  $S$  was initially solved without special combinatorial models, via exhaustive search. Soon two nice graph models were proposed. The first one, by Lysov and co-workers, places words from  $S$  in vertices of a directed graph. Two vertices  $u$  and  $v$  are connected by arc  $(u, v)$  if and only if  $\text{suf}_{k-1}(e(u)) = \text{pre}_{k-1}(e(v))$ . In such a graph, a Hamiltonian path is looked for, which corresponds to a solution of the problem, i.e., the properly ordered sequence of all words from  $S$  [30]. In the second model by Pevzner, words from  $S$  correspond to arcs and their prefixes and suffixes of length  $k - 1$  to vertices. Arcs are directed from the prefix of a word to the suffix of the same word, and the solution is an Eulerian path [36]. Figure 10.10 shows the two models in an example. It also shows a common problem in bioinformatics, ambiguity of a solution, which cannot be solved without additional information, e.g., coming from other experiments or expert knowledge.



**Fig. 10.10** Graph models for the problem of DNA sequencing by hybridization without errors. **a** A nucleotide sequence of a DNA fragment and  $k$ -mers identified as its parts, here  $k = 4$  and  $S = \{‘AGAG’, ‘AGTA’, ‘AGTC’, ‘CAGA’, ‘CAGT’, ‘GAGT’, ‘GTCA’, ‘TCAG’\}$ . **b** The graph by Lysov *et al.* constructed for  $S$ , where there are two Hamiltonian paths corresponding to two possible solutions of the problem: ‘CAGTCAGAGTA’ and ‘CAGAGTCAGTA’. **c** The graph by Pevzner constructed for  $S$  with two Eulerian paths resulting in the same two nucleotide sequences

### 10.3.3 Directed Line Graphs, De Bruijn Graphs, and Others

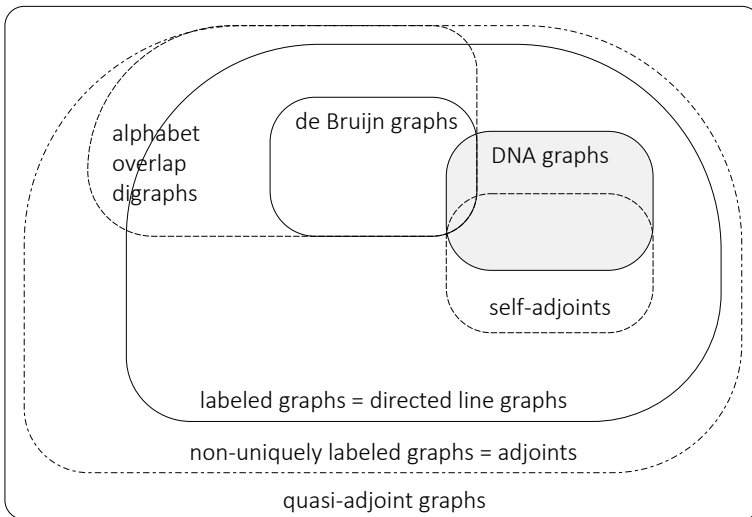
The equivalence of the two graph models became a subject of research a decade later. Why, in this case, is the transformation from the NP-hard problem of the Hamiltonian path to the polynomially solvable Eulerian path possible? The answer was given in [8], the Lysov graph is a directed line graph of the Pevzner graph constructed for the same set  $S$ , and the problems of the Hamiltonian path or cycle in directed line graphs are polynomially solvable. In [8] also a wider analysis of labeled graphs was done. The graphs of Lysov and Pevzner have labels at vertices, and the overlapping labels in Lysov graphs imply the presence of arcs (it is not the case of Pevzner graphs). Actually, only the Lysov graphs can be classified as the labeled graphs, they belong to classes  $\mathcal{L}_k^4$ . Lysov graphs are also called *DNA graphs*, especially when not restricted to errorless  $S$ . DNA graphs are vertex-induced subgraphs of de Bruijn graphs with  $\alpha = 4$ . Pevzner graphs are subgraphs of DNA graphs, thus in consequence, subgraphs of de Bruijn graphs.

Currently, a widest superclass of labeled graphs that is ‘easy’ for the Hamiltonian cycle/path problem is the class of quasi-adjoint graphs [12]. In general, such graphs cannot be labeled because sets of immediate successors of two vertices are no longer the same or disjoint for all the pairs within a graph. Figure 10.11 shows how graph classes mentioned here relate to each other.

### 10.3.4 Other Variants of DNA Sequencing

The DNA sequencing by hybridization is usually considered with errors accompanying the experiment. Then, it is also modeled as a graph problem but without the useful property of the polynomial-time solvability. Presence of any errors in an instance of the DNA sequencing problem makes the problem (i.e., the reconstruction of the original nucleotide sequence of a DNA fragment) strongly NP-hard [9]. Both methods of graph construction, by Lysov *et al.* and Pevzner, still work for  $S$  with errors, but the Hamiltonian or Eulerian path cannot be expected there. The class of DNA graphs, as presented in Fig. 10.11, covers graphs constructed according to Lysov *et al.* for all possible sets  $S$  (with or without experimental errors) that do not contain repetitions nor words of different lengths.

With next steps going beyond the original Lysov’s method, we lose ties with graphs from Fig. 10.11. If we allow overlaps of  $k$ -mers with an offset greater than 1 but constant, we still obtain a (non-uniquely) labeled graph, a vertex-induced subgraph of an alphabet overlap digraph. But for a variable offset allowed, the resulting graph is outside the class of quasi-adjoint graphs. Look at the following example, a pair of vertices ‘TGATAT’ and ‘CCATAT’, and their sets of successors { ‘GATATA’, ‘ATATTA’ } and { ‘ATATTA’, ‘CATATT’ }, respectively. The sets are not disjoint, nor equal, nor one contained in the other, thus do not match the property for quasi-adjoint graphs.



**Fig. 10.11** Relations between the classes of uniquely and non-uniquely labeled digraphs, their subclasses, and quasi-adjoint graphs [26]. DNA graphs (Lysov graphs) are used as models of DNA sequencing. All the graphs included here are polynomial-time solvable instances of the problems of searching for the Hamiltonian cycle or the Hamiltonian path

A non-classical approach to the DNA sequencing by hybridization, the isothermic DNA sequencing, produces a set of nucleotide subsequences that can be characterized by the same ‘temperature’ (melting temperature of DNA duplexes) but differing in length [7]. A graph proposed as a model for this variant, for the errorless case, can be either a directed line graph, an adjoint not being a directed line graph, or a quasi-adjoint graph not being an adjoint [10]. Therefore, this problem is also polynomially solvable.

The DNA sequencing by hybridization was replaced by a newer technique, a high-throughput automated sequencing, which is realized without an algorithmic stage. However, algorithms and models are still necessary for such sequencing data at the next level of organizing them. Although the output data from these two sequencing approaches have different scale and contain different errors, basically the processes of composing partial sequences into a final DNA fragment are very similar. Consequently, graph models of Lysov *et al.* and Pevzner are still in use, after some necessary adjustments to new circumstances (for a detailed description see, for example, [11]). One of the adjustments is the permission for non-exact overlaps of sequences in the Lysov’s model. However, this one change makes a graph not satisfying the property for quasi-adjoint graphs.

## 10.4 Conclusions

In this chapter a short review of selected applications of labeled graphs in life sciences has been given. Graph theory is a very important and useful tool in solving various problems appearing in many areas of biological sciences. Its application helped to make a progress in DNA sequencing and mass spectrometry, among others. On the other hand, problems arising in biology (especially molecular biology) are inspirations for new directions of theoretical research in graph theory (DNA graphs and molecular graphs being examples). So, the intersection of these two seemingly not very closely related areas, i.e., biology and graph theory, is a source of many interesting problems, results and inspirations for both of them.

## References

1. Akutsu, T., Nagamochi, H.: Comparison and enumeration of chemical graphs. *Comput. Struct. Biotechnol. J.* **5**(6), e201302,004 (2013)
2. Andersen, J.L., Flamm, C., Merkle, D., Stadler, P.F.: An intermediate level of abstraction for computational systems chemistry. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **375**(2109), 20160,354 (2017)
3. Badertscher, M., Korytko, A., Schulz, K.P., Madison, M., Munk, M.E., Portmann, P., Junghans, M., Fontana, P., Pretsch, E.: Assemble 2.0: a structure generator. *Chemometrics Intell. Lab. Syst.* **51**(1), 73–79 (2000)
4. Bains, W., Smith, G.: *J. Theor. Biol.* **135**, 303–307 (1988)

5. Berge, C.: *Graphs and Hypergraphs*. North-Holland Publishing Company, London (1973)
6. Biggs, N., Lloyd, E.K., Wilson, R.J.: *Graph Theory*, pp. 1736–1936. Oxford University Press, Oxford (1986)
7. Blazewicz, J., Formanowicz, P., Kasprzak, M., Markiewicz, W.: Sequencing by hybridization with isothermic oligonucleotide libraries. *Discrete Appl. Math.* **145**, 40–51 (2004)
8. Blazewicz, J., Hertz, A., Kobler, D., de Werra, D.: On some properties of DNA graphs. *Discrete Appl. Math.* **98**(1–2), 1–19 (1999)
9. Blazewicz, J., Kasprzak, M.: Complexity of DNA sequencing by hybridization. *Theor. Comput. Sci.* **290**, 1459–1473 (2003)
10. Blazewicz, J., Kasprzak, M.: Computational complexity of isothermic DNA sequencing by hybridization. *Discrete Appl. Math.* **154**, 718–729 (2006)
11. Blazewicz, J., Kasprzak, M., Kierzyńska, M., Frohberg, W., Swiercz, A., Wojciechowski, P., Zurkowski, P.: Graph algorithms for DNA sequencing—origins, current models and the future. *Eur. J. Oper. Res.* **264**, 799–812 (2018)
12. Blazewicz, J., Kasprzak, M., Leroy-Beaulieu, B., de Werra, D.: Finding Hamiltonian circuits in quasi-adjoint graphs. *Discrete Appl. Math.* **156**, 2573–2580 (2008)
13. Böcker, S., Lipták, Z., Martin, M., Pervukhin, A., Sudek, H.: DECOMP—from interpreting mass spectrometry peaks to solving the money changing problem. *Bioinformatics* **24**(4), 591–593 (2008)
14. Brown, H., Hjelmeland, L., Masinter, L.: Constructive graph labeling using double cosets. *Discrete Math.* **7**(1–2), 1–30 (1974)
15. Brown, H., Masinter, L.: *An algorithm for the construction of the graphs of organic molecules*. Stanford University (1973)
16. de Bruijn, N.: A combinatorial problem. *Proc. Koninklijke Nederlandse Akademie van Wetenschappen* **49**, 758–764 (1946)
17. Faradzev, I.: Constructive enumeration of combinatorial objects. *problèmes combinatoires et théorie des graphes* **260**, 131–135 (1978)
18. Faulon, J., Visco, D.P., Roe, D.: Enumerating molecules. *Rev. Comput. Chem.* **21**, 209 (2005)
19. Faulon, J.L.: On using graph-equivalent classes for the structure elucidation of large molecules. *J. Chem. Inform. Comput. Sci.* **32**(4), 338–348 (1992)
20. Faulon, J.L.: Isomorphism, automorphism partitioning, and canonical labeling can be solved in polynomial-time for molecular graphs. *J. Chem. Inform. Comput. Sci.* **38**(3), 432–444 (1998)
21. García-Domenech, R., Gálvez, J., de Julián-Ortiz, J.V., Pogliani, L.: Some new trends in chemical graph theory. *Chem. Rev.* **108**(3), 1127–1169 (2008)
22. Gugisch, R., Kerber, A., Kohnert, A., Laue, R., Meringer, M., Rücker, C., Wassermann, A.: Molgen 5.0, a molecular structure generator. In: *Advances in Mathematical Chemistry and Applications*, pp. 113–138. Elsevier, Amsterdam (2015)
23. Hakimi, S.L.: On realizability of a set of integers as degrees of the vertices of a linear graph. *J. Soc. Ind. Appl. Math.* **10**(3), 496–506 (1962)
24. Hao, J.: The adjoints of DNA graphs. *J. Math. Chem.* **37**, 333–346 (2005)
25. Hsieh, S.M., Hsu, C.C., Hsu, L.F.: Efficient method to perform isomorphism testing of labeled graphs. In: *International Conference on Computational Science and Its Applications*, pp. 422–431. Springer, Berlin (2006)
26. Kasprzak, M.: Classification of de Bruijn-based labeled digraphs. *Discrete Appl. Math.* **234**, 86–92 (2018)
27. Kozak, A., Glowacki, T., Formanowicz, P.: On a generalized model of labeled graphs. *Discrete Appl. Math.* **161**(13–14), 1818–1827 (2013)
28. Li, X., Zhang, H.: Embedding on alphabet overlap digraphs. *J. Math. Chem.* **47**, 62–71 (2010)
29. Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.* **25**(1), 42–65 (1982)
30. Lysov, Y., Florentiev, V., Khorlin, A., Khrapko, K., Shik, V., Mirzabekov, A.: Determination of the nucleotide sequence of DNA using hybridization with oligonucleotides. A new method. *Doklady Akademii Nauk SSSR* **303**, 1508–1511 (1988)



31. Maxam, A., Gilbert, W.: A new method for sequencing DNA. *Proc. Nat. Acad. Sci. USA* **74**, 560–564 (1977)
32. Minkin, V.I.: Glossary of terms used in theoretical organic chemistry. *Pure Appl. Chem.* **71**(10), 1919–1981 (1999)
33. Pease, A.C., Solas, D., Sullivan, E.J., Cronin, M.T., Holmes, C.P., Fodor, S.: Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proc. Nat. Acad. Sci.* **91**(11), 5022–5026 (1994)
34. Peironcely, J.E., Rojas-Chertó, M., Fichera, D., Reijmers, T., Coulier, L., Faulon, J.L., Hanke-meier, T.: Omg: open molecule generator. *J. cheminformatics* **4**(1), 21 (2012)
35. Pevac, S., Crundwell, G.: Polyá's isomer enumeration method: a unique exercise in group theory and combinatorial analysis for undergraduates. *J. Chem. Educ.* **77**(10), 1358 (2000)
36. Pevzner, P.: l-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.* **7**, 63–73 (1989)
37. Pogliani, L.: From molecular connectivity indices to semiempirical connectivity terms: recent trends in graph theoretical descriptors. *Chem. Rev.* **100**(10), 3827–3858 (2000)
38. Polyá, G., Read, R.C.: *Combinatorial enumeration of groups, graphs, and chemical compounds*. Springer Science & Business Media, Berlin (2012)
39. Read, R.C.: Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. In: *Annals of Discrete Mathematics*, vol. 2, pp. 107–120. Elsevier, Amsterdam (1978)
40. Russell, S., Norvig, P.: *Artificial intelligence: a modern approach* (2002)
41. Sanger, F., Nicklen, S., Coulson, A.: DNA sequencing with chain-terminating inhibitors. *Proc. Nat. Acad. Sci. USA* **74**, 5463–5467 (1977)
42. Southern, E.: Analyzing polynucleotide sequences. International patent application PCT/GB89/00460 (1988)
43. Sutherland, G.: Dendral—a computer program for generating and filtering chemical structures. Technical report, Department of Computer Science, Stanford University of California (1967)
44. Torán, J.: On the hardness of graph isomorphism. *SIAM J. Comput.* **33**(5), 1093–1108 (2004)