Computing and Informatics, Vol. 30, 2011, 381–410

EXPLORING COMPLEX NETWORKS WITH GRAPH INVESTIGATOR RESEARCH APPLICATION

Wojciech Czech, Witold Dzwinel

Institute of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30
30-059 Kraków, Poland
e-mail: czech@agh.edu.pl

Sławomir Goryczka

Department of Mathematics and Computer Science
Emory University
400 Dowman Drive, Atlanta, GA 30322, USA

Tomasz Arodź

Department of Computer Science
School of Engineering, Virginia Commonwealth University
401 West Main Street, Richmond, Virginia 23284-3019, USA

Arkadiusz Z. Dudek

Oncology and Transplantation Division
Medical School, University of Minnesota
420 Delaware Street S. E., Minneapolis, MN 55455, USA

Manuscript received 9 November 2009; revised 18 January 2011
Communicated by Jacek Kitowski

Abstract. This paper describes *Graph Investigator*, the application intended for analysis of complex networks. A rich set of application functions is briefly described including graph feature generation, comparison, visualization and edition. The program enables to analyze global and local structural properties of networks with the use of various descriptors derived from graph theory. Furthermore, it allows to quantify inter-graph similarity by embedding graph patterns into low-dimensional space or distance measurement based on feature vectors. The set of available graph descriptors includes over eighty statistical and algebraic measures. We present two examples of real-world networks analysis performed with *Graph Investigator*: comparison of brain vasculature with structurally similar artificial networks and analysis of vertices importance in a macaque cortical connectivity network. The third example describes tracking parameters of artificial vascular network evolving in the process of angiogenesis, modelled with the use of cellular automata.

Keywords: Complex networks, graph descriptor, graph matching, biological network, graph pattern vector

Mathematics Subject Classification 2000: 68R10, 68T10, 68-04

1 INTRODUCTION

In this work we present *Graph Investigator* application, which provides convenient and fully featured framework for network-oriented research and analysis. Computational investigation of networks plays fundamental role in various fields of contemporary basic and applied science, ranging from discrete mathematics to biology. Ubiquity of graph patterns in nature motivates the development of graph models, reflecting relations between different objects and allowing for elucidation of principles that govern complex systems [1]. Network approach to system analysis, represented by the theory of complex networks, succeeded in uncovering universal rules exhibited by plenty of social, technological and biological networks [2]. The quantitative research of graph data requires development of robust algorithms and tools, intended for graph feature extraction, graph matching and graph visualization.

The analysis of high-volume and rapidly growing datasets of biological networks, derived from molecular or cellular scale, brings deep insight into basic mechanisms of life. The molecular interaction networks epitomized by metabolic networks or protein-protein interaction networks stores huge amount of data about cell functional organization [3]. Research carried out on such networks allows for uncovering evolutionary relations between organisms [4, 5] and elucidation of local structure similarity across species [6]. Identifying fragments of networks vulnerable to single or multi-target attack and investigation of network resistance to different damages helps in designing new drugs and therapies [7, 8]. Graph-theoretical analysis of brain functional and anatomical connectivity can be helpful for diagnosis of such

diseases as schizophrenia [9, 10], epilepsy [11] or Alzheimer's disease [12]. Networks modeling vasculature or connections between neurons play significant role in understanding processes taking place in organs. Vascular networks analysis can have great impact on advances of anti-angiogenesis therapy in cancer. Comparing blood vessels in different tumors may test hypothesis about their similar growth pattern and metastatic potential. Promising application of graph matching algorithms arises as far as inter-regional variation of vasculature in tumor is considered. Quantifying relations between vascular networks in different types of tumors can also bring valuable conclusions.

The applications described above require robust tool capable of capturing various topological features of networks and allowing to perform inter-network comparisons on the basis of diverse criteria. Investigation of network topology using graph descriptors can help understand the behavior of the whole system but also evaluate models by comparing results of *in-silico* experiments with *in-vivo* ones. The structure of network encodes a number of global and local information, which can be extracted by dedicated measures and used in further analysis. Quantitative evaluation of such network properties as connectivity, symmetry, ability to transfer signals or to form node clusters can reveal qualitative features of underlying complex system. In addition, local information e.g. vertex centrality or vertex clustering coefficient enables to find subnetworks playing the most significant role in the whole system and allows for grouping functionally similar nodes. The analysis of vertex or edge-features distributions can also bring insight into system-level characteristics.

The aim of this work is to describe *Graph Investigator* – application designed to perform all the above-mentioned analytic tasks. We introduce new software system whose main features are rich functionality, platform-independence, effectiveness and modern look. Our application has strong capability to explore network data originating from real-world experiments and artificial models, and we believe it to be useful in biomedical areas.

The rest of this paper is organized as follows. In Section 2 we review different network-analytic tools including applications and libraries and compare them with our framework, explaining reasons for starting this work. Section 3 provides details on computational methods and theory behind *Graph Investigator*, presenting available graph descriptors divided into two groups: general and algebraic. In Section 4 we describe features of our application, while Section 5 presents three typical use cases involving biomedical data sets: brain vascular network, macaque cortical connectivity network and vascular network in presence of tumor. Then, in Section 6 hardware and system requirements of *Graph Investigator* are specified. Section 7 explains how the program can be downloaded. Finally, we discuss conclusions.

2 RELATED WORK

Networks analysis software developed so far provide a variety of methods to investigate large datasets. However, their practical use is often limited to the networks of

specific type, e.g. social networks. Consequently, the set of available descriptors is domain-biased. Existing applications share many features, for instance most them possess a visualization module. Besides, the set of available graph descriptors is often modest, containing basic measures and lacking more specific ones.

A common example of large network analysis tool is *Pajek* [14], a Windows application capable of visualizing graphs and performing statistical analysis of their topology. Despite advantages such as stability and maturity, platform dependence and closed code decrease flexibility of this application. The *NetworkX* Python package [13] provides a productive framework for the study of networks but requires programming skills. The tools intended for statistical social networks analysis and modelling are represented by *StOCNET* [15] or *Visone* [16] applications. *StOCNET*, an open source Windows program, does not provide analytic tools from spectral graph theory and its application to networks other than social is limited. *Visone* is a closed source Java application with strong visualization capabilities. It allows for computation of vertex metrics but more advanced features such as forming feature vectors and embedding graph into pattern space are not available. From the perspective of spectral graph theory, investigation of networks structure can be carried out using *Spectral net* [17] program. Unfortunately, this .NET application seems to be no longer maintained. Commercial graph exploration software is exemplified by *NetMiner* [18] or *orgnet* [19] frameworks.

Our motivation for starting implementation of *Graph Investigator* was to create an application focused on computing rich set of both statistical and spectral graph descriptors and capable of network feature generation for comparing real-world networks with some model network types. We decided to use *Java* programming language to provide platform independence and to take advantage of robust graph libraries as *JUNG* [20] and *Prefuse* [21]. The former provides number of algorithmic tools while the latter is an advanced visualization framework.

3 COMPUTATIONAL METHODS AND THEORY

A wide variety of graph-analytic metrics available in *Graph Investigator* application enables to perform an in-depth study of graph structure. In this section we briefly present a subset of available descriptors (the rest being described in Appendix), giving some explanations of their purpose. Categorization of the metrics can be carried out in several ways. Here, we decided to make a distinction between descriptors computed as permutation invariant function of local graph properties, and vector descriptors obtained from spectral graph theory. The origins of the presented metrics are diverse. Some of them, as the Wiener index, arise from the topological studies of molecules [29, 32] while other derive from the analysis of social networks.

In this section we also describe *spectral embedding* – graph visualization method [38, 39] originating from spectral graph theory that can deliver much valuable information about graph structure. At the end of the section we present some general remarks concerning graph matching based on feature vectors.

3.1 General Graph Descriptors

Let us recall some basic notions from graph theory that will be used further in this text. Graph G is defined as ordered pair $G = (V, E)$, where V is a set of vertices and E is a set of edges. An edge $e_{uv} = \{u, v\} \in E$ is an unordered pair of vertices. The vertices model objects while edges model relations between them. Two vertices u and v are adjacent ($u \sim v$) if they are joined by an edge. A sequence of vertices such that from each vertex there is an edge to the next vertex in the sequence is called a path. The length of the path is the number of edges it contains. The distance between vertex u and v, denoted by $d(u, v)$, is a length of the shortest path between u and v. If a path between u and v does not exist then $d(u, v) = \infty$. The set of vertices adjacent to vertex v, denoted by N_v, is called its neighborhood. The degree (or valence) k_v of vertex v is a number of edges which join v with its neighbors. Graph G is connected if a path exists between each pair of vertices.

Clustering Coefficient. This graph descriptor measures neighborhood connectivity used to indicate small-worldliness of a network [27]. The clustering coefficient of vertex v defined as follows:

$$C(v) = \frac{2|\{e_{ij}\}|}{k_v(k_v - 1)}, \tag{1}$$

where $i, j \in N_v$ and $e_{ij} = \{i, j\} \in E$, is a ratio of number of connections between neighbors of vertex v (denoted by $|\{e_{ij}\}|$) to number of links that could possibly exist between them, i.e., $k_v(k_v - 1)/2$. Clustering coefficient quantifies local topology of a graph, reflecting how close the neighborhood of a given vertex is to form a complete graph. The clustering coefficient for a graph G with n vertices is an average of clustering coefficients for each vertex.

$$C(G) = \frac{1}{n} \sum_{v \in V} C(v) \tag{2}$$

The above formula is undefined for vertices of degree 1 or 0. In these cases $C(v)$ is usually set to 0, however, as pointed out in [28], such procedure may lead to biased assessments of neighborhood clustering when undefined values dominate average. Therefore, additionally we provide different graph clustering coefficient computed using vertices with more than one neighbor [28]:

$$C'(G) = \frac{1}{n'} \sum_{v \in V : k_v > 1} C(v), \tag{3}$$

where n' is the number of vertices of valence greater than 1.

Wiener Index. Wiener index, proposed in [29], is a sum of lengths of the shortest paths between every pair of vertices in a given graph G.

$$W(G) = \frac{1}{2}\sum_{u\in V}\sum_{v\in V} d(u,v) \tag{4}$$

It is used in chemistry as a topological index of molecule, reflecting its branching and correlated with its van der Waals surface. Wiener index reaches minimal value $\frac{1}{2}n(n-1)$ for a complete graph and maximal value $\frac{1}{6}(n^3-n)$ for a path graph, where n is the number of vertices. Using the latter formula we can normalize Wiener index as follows

$$W_n(G) = \frac{6W(G)}{n^3-n}, \tag{5}$$

obtaining $W_n(G) \in [0,1]$.

Subgraph Count. The k^{th}-order Subgraph Count for a graph G, ${}^kSC(G)$, is a number of connected subgraphs with k edges.

$${}^kSC(G) = |\{H = (V_H, E_H) \subset G\colon |E_H| = k \wedge diam(H) < +\infty\}| \tag{6}$$

This descriptor shows how many specified subgraphs with k edges occur in the graph. It can be normalized as follows

$${}^kSC_n(G) = \frac{{}^kSC(G)}{{}^kSC(K)}, \tag{7}$$

where K is a complete graph (clique) with n vertices. The common descriptors, derived from generic SC descriptors are Platt Index [30] ($l = 2$) and Gordon-Scantleburry Index [31] ($l = 3$). They reach their maximal values for cliques.

Betweenness Centrality. Centrality is a measure of relative importance of graph vertex according to given criteria. Betweenness measures are a kind of centrality measures used often in the analysis of social or citation networks. They tend to evaluate the influence of each vertex on spreading information over the graph. Shortest-path betweenness is a widely used centrality measure defined as a fraction of the shortest paths between pairs of vertices in a graph that pass through given vertex [33, 34], i.e.,

$$BC(v) = \sum_{s\neq v\neq t\in V s\neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{8}$$

where $\sigma_{st}(v)$ is the number of the shortest paths from s to t which pass through v and σ_{st} denotes the total number of the shortest paths from s to t. It measures to what extent a given vertex is needed by other vertices to transfer information through

the shortest paths. The betweenness centrality for a graph G with n vertices is an average of betweenness centrality for each vertex.

$$BC(G) = \frac{1}{n} \sum_{v \in V} BC(v) \tag{9}$$

Random Walks Betweenness Centrality. The shortest path betweenness described above aims at evaluating the influence of each vertex on spreading information over network through geodesic routes. This model assumes that information does not move through other non-optimum paths, what in many cases is not true. In [35], Newman proposed a general betweenness measure, which takes into account all paths between any two nodes yet giving greater weight to shorter paths. This betweenness can be defined either by an analogy to current flow in an electrical network or by random walks. These two definitions are equivalent. In the first one, we treat a graph as an electrical circuit with unit resistances on edges and average the current flowing through given vertex out, involving all pairs of current-in and current-out. In the latter definition, one calculates the mean number of passages the random walker travelling between any pair of nodes makes through a given vertex. The random walks betweenness centrality $RWB(G)$ is an average of random walks betweenness centrality for each vertex. The rest of available descriptors is described in Appendix.

3.2 Algebraic Graph Descriptors

Spectral graph theory finds correspondences between structural properties of networks and algebraic properties of matrices representing graphs. The most common matrix representation for graph $G = (E, V)$ is adjacency matrix defined as follows

$$\mathbf{A}_{u,v} = \begin{cases} 1 & \text{if } \{u, v\} \in E \\ 0 & \text{if } \{u, v\} \notin E, \end{cases} \tag{10}$$

nevertheless during generation of algebraic graph feature vectors we usually use Laplace matrix or one of its variants such as normalized Laplacian. The Laplacian, defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ denotes diagonal matrix of vertex degrees, is well suited to graph analysis as it possesses only nonnegative eigenvalues [38]. This helps to order eigenvectors and use them to generate features of the graph.

Heat kernel – the fundamental solution of heat equation associated with the Laplacian (see Equations (11) and (12), ϕ_i denotes eigenvector associated with an eigenvalue λ_i) allows for construction of valuable graph metrics that additionally can be scaled by the time parameter [49, 50]. This enables to navigate between local properties (low values of t) and global properties (high values of t) of graph.

$$\frac{\partial h_t}{\partial t} = -\mathbf{L} h_t \tag{11}$$

$$h_t = \exp(-\mathbf{L}t) = \sum_{i=1}^{|V|} \exp(-\lambda_i t)\phi_i \phi_i^T \tag{12}$$

Eigendecomposition and Singular Value Decomposition (SVD) of graph matrices deliver many valuable information about graph topological properties. For instance, the sum of 3$^{\text{rd}}$ powers of adjacency matrix eigenvalues can be used to compute a number of triangles in graph [38]. The spectrum of Laplacian helps establish a number of spanning trees, a number of connected components or estimate isoperimetric number of a graph [36]. Therefore the results of algebraic decomposition form rich basis to generate graph descriptors. *Graph Investigator* allows for compuation of various graph descriptors based on eigendecomposition of Laplace matrix or heat kernel matrix (see Appendix).

3.3 Graph Spectral Embedding

Eigendecomposition of Laplacian matrix can be employed to visualize graphs. In matrix representation we assume that vertices are ordered, therefore successive elements of selected eigenvector can be treated as scalars assigned to nodes. Taking into account two eigenvectors, for each vertex we obtain 2D coordinates that can be used to place it into 2D space. Next, we connect adjacent vertices with a line. Using eigenvectors corresponding to two smallest non-zero eigenvalues, we can get very useful visualization of graph, capable of amplifying its important features and revealing hidden structural patterns [38, 39]. For planar graphs particularly, such an embedding provides good coordinates for nodes.

4 PROGRAM DESCRIPTION

Graph Investigator is a program written in Java, available upon request (*mailto:* `czech@agh.edu.pl`). It was designed to be extended easily, hence new functionality e.g. new graph descriptors can be added without much effort. Provided that Java Runtime Environment is installed, *Graph Investigator* can run on many different platforms as Linux, Solaris or Windows.

The functionality of *Graph Investigator* is described below. First we present graph data formats that can be used with our application. Then the basic analytic tools are described. Next we report available visualization modules and finally we show how to perform statistical analysis of vertex descriptors.

4.1 Input/Output

Graph Investigator can work with real-world or artificial networks. The latter can be generated using the following models: Barabási-Albert [45], Eppstein power law [46], Erdős-Rényi [54], Kleinberg small-world, Watts small-world, scale-free, clique model, path model, star model and balanced tree model. Artificial networks are frequently

used for comparison with real-world data. Such analysis can reveal rules governing creation of real-world networks. *Graph Investigator* accepts the following graph input formats: simple list of edges (*.edgelist), Graph ML (*.xml), Amira skeleton (*.am) [47], Pajek (*.net), edgelist (*.edge), adjacency matrix (*.mtrx) and binary graph [55]. A detailed description of these file formats is available in application documentation. A user can also save graph with the use of any of the above-mentioned formats and perform command line format conversion for large sets of input data. Uploaded networks are presented in the table (see Figure 1) with tooltips providing general information such as number of edges and number of vertices.

4.2 Graph Descriptors

The analysis of network structure can be performed with the help of 87 scalar and 9 vector descriptors, presented in Section 3 and Appendix. *Graph Investigator* provides two modes of invariants computation: graph descriptors and vertex/edge descriptors. Scalar metrics can be aggregated into one feature vector and used in this form in further analysis (e.g. using *Principal Component Analysis*). Computation of vertex-attribute-based statistical moments such as mean, standard deviation, skewness and kurtosis is also provided. *Graph Investigator* enables to draw histograms of edge/vertex/path descriptors. As shown in Figure 1, graph descriptors can be presented in a table while edge/vertex descriptors in a edge/vertex tooltip on a graph picture. Values of vertex descriptors can be also shown in graph visualization window using predefined colormap (see Figure 1). Spectral mode enables to calculate algebraic decompositions (Eigendecomposition, SVD) of several graph matrices such as adjacency matrix, Laplacian or normalized Laplacian. It also provides comparison of spectral density functions for selected graphs. Additionally, application allows for computation of descriptors based on heat kernel matrix [49]. Graph metrics can be saved in CSV format (*Comma Separated Values*).

4.3 Visualization

Graph Investigator provides three modes of graph visualization: with force simulation (see Figure 2 b)), using *B matrix* of graph [40] (Figure 1, B matrix internal window) and using spectral embedding [38, 39] (see Figure 5). The parameters of force simulation can be adjusted to obtain different graph views. The vertex or edge descriptors are visualized as tooltips or colors on graph drawing. The graph can be also edited by hand (edge or vertex addition/deletion) or using visual editor. The results of graph visualization can be saved in the following graphic file formats: pdf, svg, jpg, png.

4.4 Other Features

We provide unique feature for clustering visualization. The set of graphs representing certain real-world objects can be embedded into lower-dimensional space, where

datapoints are visualized as mini-photos (photos representing objects are associated to data as described in application documentation). The available dimensionality reduction methods include *Principal Component Analysis* (PCA), *Locally Linear Embedding* (LLE) [41], *Isomaps* [42], *Kernel PCA* [43], *Locality-Preserved Maximum Information Projection* (LPMIP) [44] and additionally the method of particles in visual clustering [73].

Some additional features include: extracting greatest connected component, extracting all connected components, testing graph descriptors using random edit operations, adjacency matrix matching via *Singular Value Decomposition*.

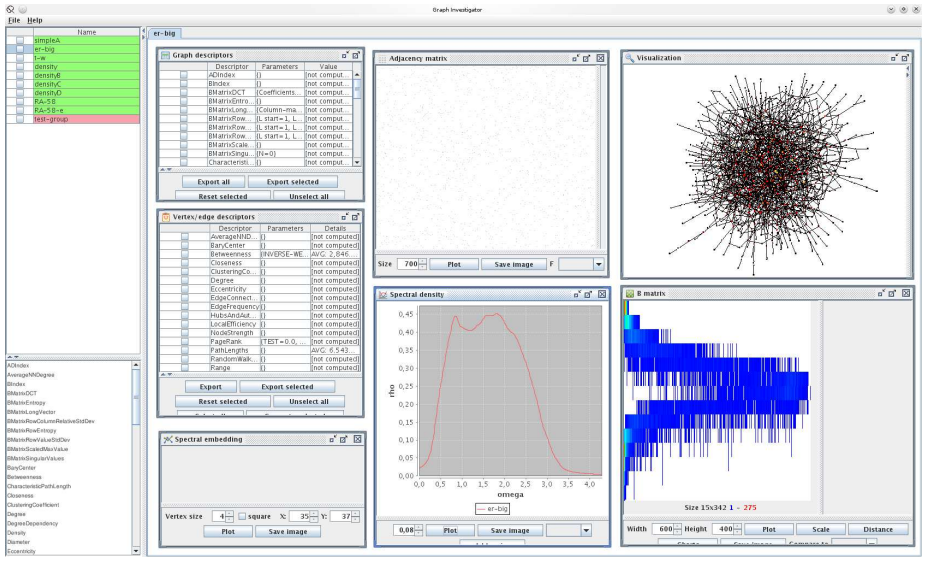


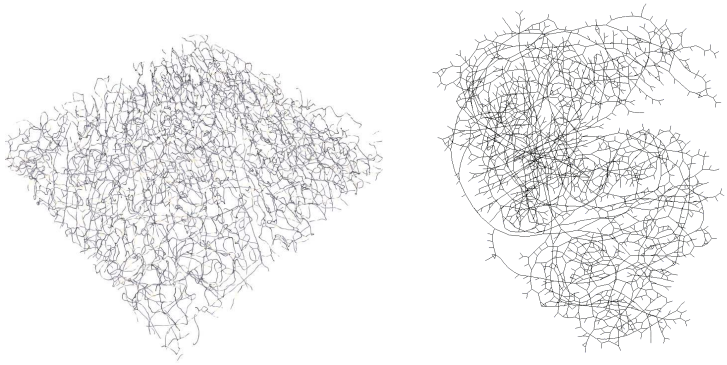
Fig. 1. The main window of *Graph Investigator*. Left – graph panel with a table containing loaded single-graph and multi-graph datasets. The main panel (right) shows internal frames that allow for graph analysis and visualization.

5 USE CASES

5.1 Brain Vascular Network

The complex network discussed in this section represents brain cerebral microvascularization, derived from 3D images acquired by confocal microscopy [22]. The moisac of 3D images can be transformed into 3D skeleton using the algorithm introduced in [22]. The skeleton, described by the set of points and curves, is read by *Graph Investigator* and converted into graph with 2 206 vertices and 2 983 edges. Spatial visualization of this brain vascular network is depicted in Figure 2 a). For the sake of simplicity, in further analysis, we take into account the greatest connected

component (2 131 vertices and 2 846 edges) of obtained network. The Figure 2 b) presents visualization of this component using force simulation (*Graph Investiagtor*).



a) b)

Fig. 2. Visualization of brain vascular network using *Amira* a) and *Graph Investigator* with force simulation b). Picture a) reflects real locations of vessels in space, whereas b) emphasizes topological features with endings (forks) and long paths better visible.

In Figure 3 the histogram of vertex degrees in vascular network is depicted. The network has nearly uniform distribution of vertex degrees, nevertheless vertices with number of neighbors greater than 3 also occur. The vertices of valence 3 dominate, which is a typical feature of river networks or tree branches, that possess three-way junctions (forks) most likely. A relatively high number of degree 1 vertices appears due to boundary cut of images from confocal microscopy (see Figure 2 a)).

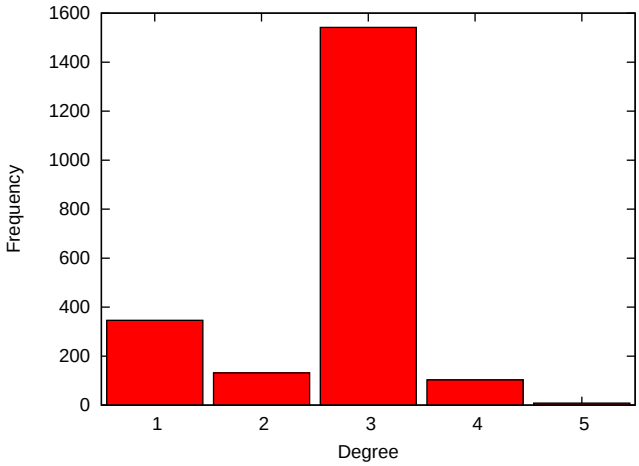
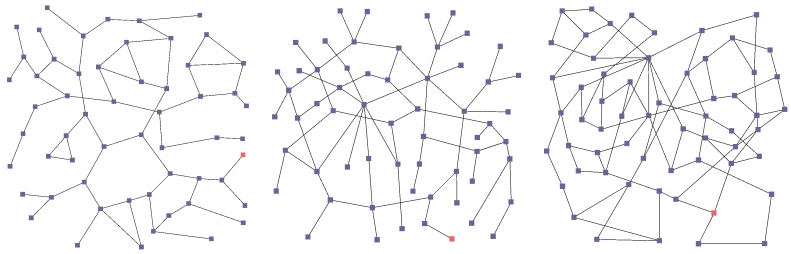


Fig. 3. Brain vascular network degree histogram

We compare the brain vascular network with two artificial networks of similar size and type using subset of descriptors available in *Graph Investigator*. Our goal is to determine if its structure is random, not revealing any patterns, or to uncover the features that make vascular network different from resembling artificial networks. First random graph was generated using Erdős-Rényi model [54] in which an edge is established between each pair of vertices with equal probability r, being the parameter of the model. The density of Erdős-Rényi graph depends on r. An example of such a graph for $r = 0.01$ and 60 vertices is depicted in Figure 4 b). Graphs obtained using Erdős-Rényi model can be unconnected. In this case we add extra edges to get connected network. The second model allows to generate irregular graphs [55], in which average valence is bounded by parameter k. The graph is built on the basis of fixed valence random graph by moving some part of edges to new locations. Here we decided to replace 10% of edges selected with uniform probability. After indicating edges to move, network vertices are permuted so that each vertex receives new index i. For a given edge new endpoint i is chosen using probability function $\alpha \exp(-\beta i)$, where α and β depend on graph size and are normalized as described in [55]. An example of irregular bounded valence graph is depicted in Figure 4 c) ($k = 3$).



a) b) c)

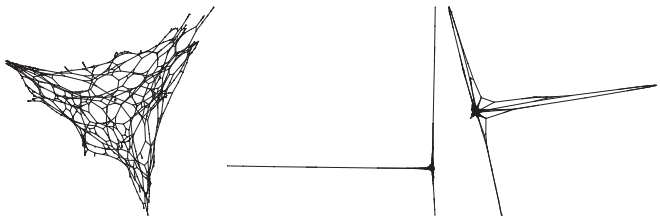
Fig. 4. Examples of graphs from dataset used to test proximity of artificial and vascular networks: a) fragment of brain vascular network (57 vertices), b) Erdős-Rényi graph ($r = 0.01$, 60 vertices) c) irregular bounded valence graph ($k = 3$, 60 vertices)

A subnetwork consisting of 1 090 vertices and 1 438 edges was extracted from brain vascular network by random selection of node and following *breadth-first* traversal with depth limited to 15. Then, we created two networks of similar size and density using Erdős-Rényi ($r = 0.001$) and irregular bounded valence models ($k = 3$). The set of graph descriptors computed for three networks is presented in Table 1.

We assumed that the size and average vertex degree are similar for all three networks. As presented in Table 1 some descriptors differ slightly while the rest capture features that separate analyzed graphs. For instance, normalized information of vertex degrees (see Appendix) is similar for all graphs. Moreover, relative differences in Randić connectivity index are also minor. It reflects that general con-

nectivity and complexity understood as number of cliques, branches etc., is similar for three considered networks (vascular network has slightly smaller structural complexity). On the other hand, values of B index, normalized Wiener index, M_1 Zagreb index, normalized Platt index and standard deviation of degree distinguish vascular network from artificial ones. These descriptors are similar for artificial networks whereas vascular network is characterized by values approximately two times larger or smaller. This indicates larger branching of vascular network, which seems to have more forks and bifurcations. The distribution of vertex degrees in vascular network is less wide than in case of artificial graphs. As far as distribution of clustering coefficients and ${}^mM_1$ Zagreb index are considered, vascular network is more similar to Erdős-Rényi graph than to irregular bounded valence graph. Yet, in terms of two statistical metrics computed on the basis of vertices betweenness centrality (standard deviation and kurtosis) vascular network is closer to irregular bounded valence graph.

To complete the comparison of the networks, we present their spectral embeddings (see Figure 5). The pictures of three graphs differ significantly. The vascular network has good embedding with triangle-like spatial distribution of vertices, whereas two other ones have great part of vertices placed in one central area. The only common characteristic is three-directional stretching of vertex coordinates. Vascular network is more two-dimensional in nature than two other ones (none of them is planar).



a) b) c)

Fig. 5. Visualization of three networks using spectral embedding (eigenvectors associated with two smallest nonzero eigenvalues of Laplacian): a) brain vascular network (1 090 vertices, 1 438), b) connected Erdős-Rényi network ($r = 0.001$, 1 028 vertices, 1 582 edges), c) irregular bounded valence network ($k = 3$, 997 vertices, 1 500 edges)

Our next step is to divide large vascular network into smaller subgraphs (possibly overlapping in part) and to try to measure to what extent such a group of graphs is structurally close to clusters of artificial networks. In this experiment we selected two groups of random graphs from the database described in [55]: the set of 20 connected Erdős-Rényi graphs (60 vertices, $r = 0.01$, denoted by ER) and the set of 20 irregular bounded valence graph (mean valence 3, denoted by BV). The group of 19 graphs was generated on the basis of large brain vascular network by random selection of core node and *breadth-first* traversal with limited depth. Only graphs of the order greater than 50 were considered. No more than 30 vertices may overlap

Descriptor	Network		
	Vascular	Erdős-Rényi	Bounded valence
Number of vertices	1 090	1 028	997
Number of edges	1 438	1 582	1 500
Diameter	30	15	18
Avg of Degree	2.639	3.078	3.01
StdDev of Degree	0.827	1.552	1.612
Kurtosis of Degree	0.411	0.260	40.699
Skewness of Degree	−0.958	0.711	5.588
M1 Zagreb index	8 334	12 212	11 616
MM1 Zagreb index	293.04	266.96	163.23
B index	0.195	0.497	0.426
Avg of Clustering Coefficient	0.222	0.153	0.033
StdDev of Clustering Coefficient	0.381	0.358	0.177
Kurtosis of Clustering Coefficient	0.218	1.796	25.537
Skewness of Clustering Coefficient	1.406	1.943	5.224
Normalized Wiener index	0.013	0.006	0.007
Avg of Betweenness Centrality	7 163.1	2 846.6	3 205.6
StdDev of Betweenness Centrality	10 068.7	3 069.0	6 321.8
Kurtosis of Betweenness Centrality	22.208	3.863	60.535
Skewness of Betweenness Centrality	3.698	1.766	6.816
Avg of RW Betweenness Centrality	0.0243	0.0123	0.0153
StdDev of RW Betweenness Centrality	0.0167	0.0076	0.0128
Normalized information of vertex degrees	0.0004	0.0005	0.0005
Normalized Platt index	4.2e−6	8.4e−6	8.7e−6
Randics connectivity index	526.3	481.6	477.8
Normalized Gordon-Scantterburry index	6.5e−9	2.3e−8	2.9e−8

Table 1. Descriptors computed for three networks: vascular, Erdős-Rényi ($r = 0.001$) and irregular bounded valence ($k = 3$)

between two different instances. This set of vascular network fragments is denoted by FV. In the Figure 4, three selected instances (one for each group) are presented. The averages of mean vertex degrees are as follows: FV 2.37, ER 2.32, BV 2.99.

On the basis of computed graph descriptors, we created multi-dimensional feature vectors to analyze structural proximity between groups of networks. Then, the pattern vectors were embedded into 2D space using *PCA* (*Principal Component Analysis*). We tested a number of combinations of descriptors, each time performing dimension reduction and evaluation of clusters separation using two validation indices – *Davies-Bouldin index* and *C index* (in 2D). The best result for 4D feature vector composed of general graph descriptors: standard deviation of degree, standard deviation of clustering coefficient, normalized Platt index and normalized Wiener index (see Section 3.1) is presented in Figure 6 a). An example of 2D em-

bedding using 2D spectral graph descriptor is depicted in Figure 6 b). This feature vector consists of mean value and standard deviation of graph Laplacian matrix spectrum (see Section 3.2). As we pointed out earlier (see Table 1), standard deviation of degree separates vascular networks from Erdős-Rényi and irregular bounded valence graphs, while standard deviation of clustering coefficient increases the distance between irregular bounded valence network and the rest.

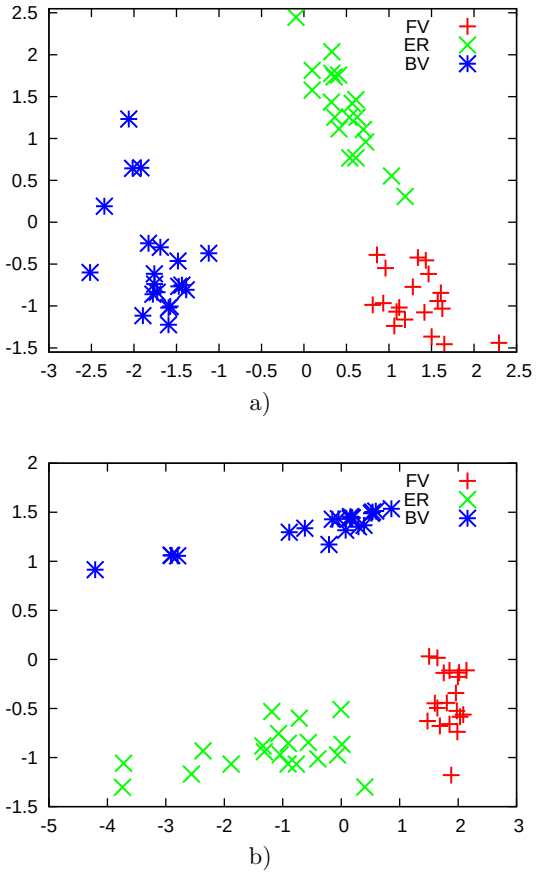


Fig. 6. Three groups of brain vascular networks embedded into 2D space (PCA) using two feature vectors: a) 4D vector composed of standard deviation of degree, standard deviation of clustering coefficient, normalized Platt index and normalized Wiener index, b) 2D vector formed of mean value and standard deviation of Laplacian matrix eigenvalues

Brain vascular network has specific structure distinguishing it from random graphs created on the basis of artificial models. The distribution of its vertex degrees is nearly uniform, with small variance. We observe relatively high clustering

coefficients, branching factors and average centralities of nodes. In case of two betweenness centrality measures, the network mentioned obtained greatest average values that indicate its efficiency in transporting blood both through shortest and random paths. These topological features are correlated with circulatory system function of delivering chemicals to tissues and are apparently the result of evolutionary optimization.

5.2 Macaque Cortical Connectivity Network

In the previous section we described how *Graph Investigator* can be used to compare networks by computing of general descriptors and forming feature vectors. Here, we present an example of local structure analysis with vertex descriptors.

The network investigated in this section reproduces macaque cortical connectivity within one brain hemisphere [23, 24, 25]. This dataset can be obtained from CoCoMac database [56, 57]. The network has 95 vertices representing primate brain areas and 1 522 edges reflecting their logical interrelations, acquired from experimental tracing studies (see Figure 7).

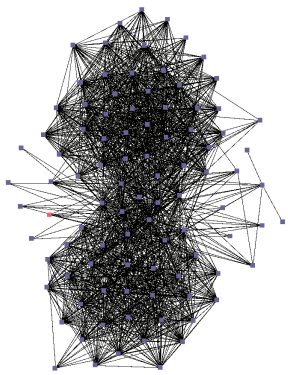


Fig. 7. Macaque brain cortical connectivity network (95 vertices, 1 552 edges).

We compute four types of descriptors assigned to vertices: degree, betweenness centrality, random walks betweenness centrality and clustering coefficient (see Section 3.1) to evaluate relevance of vertices in spreading information over the network, and quantify its local topology. The histograms, depicted in Figure 8, reveal non-uniform distribution of these metrics. As shown in Figure 8 a), vertices with valence between 25 and 45 dominate. The distribution of betweenness centrality (Figure 8 b)) is power-law-like with a great part of values lying below 0.02. In case of degree and betweenness centrality measures (8 a), 8 b), 8 c)), the groups of vertices with highest values separate well from the rest. In Table 2 we show brain areas that obtained the highest ranks in terms of three considered descriptors.

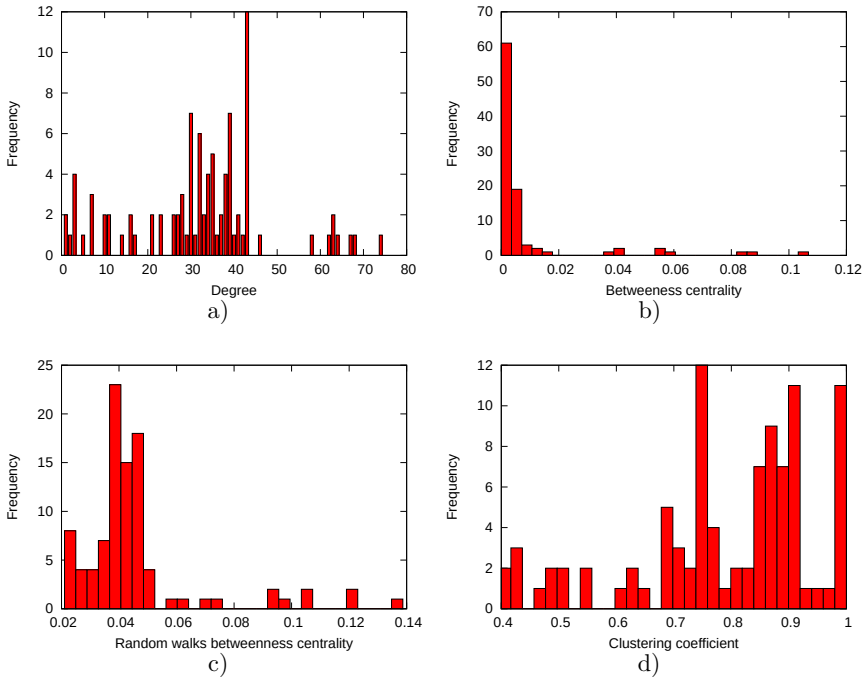


Fig. 8. Macaque brain cortical connectivity network – frequencies of vertex descriptors: a) degree, b) betweenness centrality (30 bins), c) random walks betweenness centrality (30 bins), d) clustering coefficient (30 bins)

The middle temporal area (see Figure 9) involved in motion perception and depth discrimination [60] has the greatest connectivity and centrality, as it constitutes a source of visual input for other areas. An interesting observation can be made as far as 7 b) (Brodmann area 7) and MSTdp (medial superior temporal area) areas are considered. Taking into account vertex degrees exclusively, the second highest rank area is lateral subdivision of ventral intraparietal (sensitivity to the speed and direction of motion [61]). Nevertheless, using random walks betweenness centrality descriptor, which reflects vertex importance generalized over the whole network, the second position is occupied by 7 b) – the area playing role in locating objects in space. Also betweenness centrality ranks 7 b) higher than simple degree descriptor. A similar pattern can be observed for MSTdp area, that acquired greater importance using centrality measures. This area is involved in the analysis of optic flow [62].

The three descriptors considered are strongly correlated (see Table 3) but they bring information from different scales (local to global), what results in ranks reorderings. The brain networks both in micro and macro scale are often analyzed using *small-world* metrics such as clustering coefficient and the mean shortest path [58].

Descriptor					
Degree		BC[a]		RWBC[b]	
Value	Area	Value	Area	Value	Area
74	MT[c]	**0.107**	MT	**0.139**	MT
68	VIPl[d]	**0.086**	46	**0.122**	7b
67	LIPv[e]	**0.083**	7b	**0.120**	MSTdp
64	7a[f]	**0.059**	VIPl	**0.107**	VIPl
63	7b[g]	**0.057**	MSTdp	**0.106**	LIPv
63	46[h]	**0.057**	LIPv	**0.098**	46
62	LIPd[i]	**0.042**	Id	**0.095**	7a
58	MSTdp	**0.039**	7a	**0.093**	LIPd
46	VIPm[k]	**0.036**	LIPd	0.074	VIPm

[a] betweenness centrality
[b] random walks betweenness centrality
[c] middle temporal area
[d] ventral intraparietal, lateral subdivision of VIP
[e] lateral intraparietal (ventral)
[f] visual area 7a
[g] Brodmann area 7
[h] visual area 46
[i] lateral intraparietal (dorsal)
[j] medial superior temporal area
[k] medial subdivision of VIP (ventral intraparietal)

Table 2. Highest values of three vertex descriptors: degree, betweenness centrality, random walks betweenness centrality and respective brain areas. The labels of brain areas are available in CoCoMac dataset. Decoding labels into full names was based on information from [59].

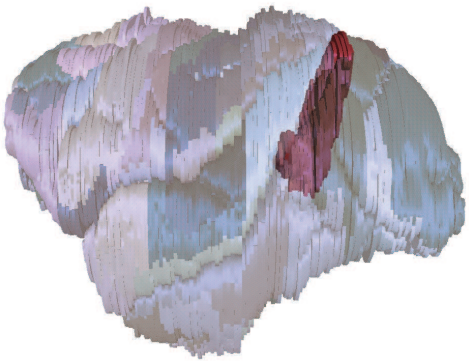


Fig. 9. Middle temporal area visualized using CoCoMac-Paxinos-3D viewer

The histogram depicted in Figure 8 d) shows that brain areas of macaque are highly clustered.

X \ Y	Degree	BC	RWBC
Degree	1	0.712	0.782
BC	0.712	1	0.861
RWBC	0.782	0.861	1

Table 3. Kendall correlation coefficients between three vertex descriptors: degree, betweenness centrality (BC) and random walks betweenness centrality (RWBC)

As shown in Table 2, computation of different topological descriptors can provide much information about functional organization of brain networks. Particularly, betweenness centrality measures reflecting influence of selected brain areas on spreading information over network captures subtle structural properties that may be more useful than simple measures like degree.

5.3 Blood Vessels in Tumor

The use case presented in this section is based on simulation data obtained from cellular automata model of tumor-induced angiogenesis [70]. We consider four sets of evolving vascular networks obtained for different parameters of the model and present how the properties of these networks change during simulation.

The model that generates the described networks consists of two interacting parts: a transportation network (blood vessels) and consuming environment (tissue) [70, 71]. The topology of the network changes in response to distribution of TAFs (Tumour Angiogenesis Factors) epitomized by VEGF (Vascular Endothelial Growth Factor) [72], which are produced by starving tumor cells in order to stimulate vessels growth. In turn, the vascular network delivers oxygen and nutrients to the tissue forming a gradient which affects tumor cells. The tissue is modelled by a mesh of cellular automata while the transportation network by a graph of cellular automata built over the CA mesh. The TAF concentration exceeding certain threshold activates vessels to create sprouts that develop towards tumor tissue. The vessel states evolve from immature to mature. Vessel maturity reflects its ability to transport blood and form new branches.

The model is characterized by a number of parameters governing rules of vascular network evolution. In this work we focus on three types of parameters that control branching: TAF threshold T_c, baseline branching probability P_b and level threshold multipliers that provide a functional relationship between TAF concentration and branching probability. Provided that TAF threshold in certain location is exceeded, the relevant vascular cell can start branching with certain probability P_b. This probability can be adjusted using two pairs of parameters: threshold T_1, multiplier m_1 and threshold T_2, multiplier m_2. If $T_c > T_1$ then P_b is multiplied by m_1 and if $T_c > T_2$ then branching probability is set to $m_2 P_b$. The parameters above

allows for increasing branching probabilities in case of high TAF concentrations, so that branching activation function has more than one step (see Figure 10).

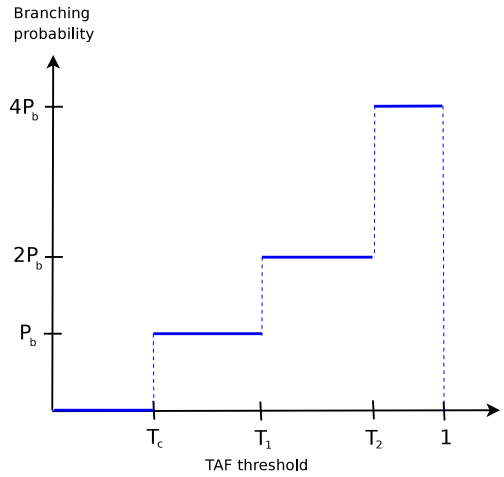


Fig. 10. Example of branching probability multipliers ($m_1 = 2$, $m_2 = 4$)

In creating angiogenesis models, an important goal is to measure how values of parameters influence growth of vascular network. Graph descriptors are a main tool in analyzing the changing topology of the growing network. We run 4 simulations with different parameters (see Table 4). Next, for each simulation step we compute graph descriptors measuring local structural properties of a generated network (mean degree and clustering coefficient). As shown in Figure 11 a), after 40 steps the sizes of networks start to grow exponentially with different rates influenced by branching probabilities. The fastest growth is observed for set02, characterized by highest branching probability for immature (young) cells. Comparing it to the slowest growth for set04, which has the same T_{cm} value (TAF threshold for mature vessels), similar T_{ci} (TAF threshold for immature vessels) and parallel multipliers for same thresholds it seems that among presented parameters P_{bi} (baseline branching probability) affects growing rate to the greatest extent.

In Figure 11 we present graph size dependencies of several metrics. Our aim is to investigate network evolution without explicit time reference and to focus on observing structural differences for networks of the same size. We investigate distributions of two local vertex descriptors: clustering coefficient and degree (see Figure 11 b), c), d)). We observe that clustering coefficient curves for set02 and set04 have similar shape with decrease at the beginning (graph sizes 40–100), increase in the middle (characteristic point at 120 where they separate from the rest) and decrease in the final phase. The decrease of clustering coefficient at the beginning reflects fast path-like growth of vascular networks towards tumor (number of vertices of degree 2 increases, minor branching rate). For degree charts similar separation (set02 and

set04 vs. set01 and set03) is observed (characteristic point at 120). It seems that TAF threshold for immature vessels is the discrimination factor here, as its possesses similar value for set02 and set04 (0.4 and 0.5 respectively) and different value for set01 and set03 (0.1). Relatively high values of T_{ci} for set02 and set04 prevent young immature cells (dominating at the beginning of simulation) from forming branches. Nevertheless, a few sprouts created from mature cells can develop towards tumor, extending vascular network size (graph sizes 40–100). Therefore after passing through *maturity* age (approximate graph size 100), the number of *mature* vascular cells increases rapidly, resulting in high branching rate (100–1 000). The growth of sets 01 and 03 is more stable, however for all four sets we observe decrease of clustering coefficient in the final phase. This effect is connected with reaching boundaries of the tissue mesh and periodic boundary conditions.

The analysis illustrates how inspecting single network properties allows for gaining insight into the processes governing the changing topology during network growth. Comparing series of graphs with the use of descriptors allows for revealing network phase transitions and understating how the parameters affect network structure. Also, the results can be used to select the most relevant parameters of the model.

	set 1	set 2	set 3	set 4
TAF threshold [a](*mature*) T_{cm}	0.01	0.01	0.01	0.01
TAF threshold (*immature*) T_{ci}	0.1	0.4	0.1	0.5
Branch probability (*mature*) P_{bm}	0.02	0.01	0.01	0.01
Branch probability (*immature*) P_{bi}	0.02	0.08	0.02	0.02
Level 1 threshold	0.6 multi 1	0.6 multi 2	0.6 multi 2	0.6 multi 2
Level 2 threshold	0.8 multi 1	0.8 multi 4	0.8 multi 6	0.8 multi 6

[a] TAF concentration is a real value between 0 and 1

Table 4. The parameters of angiogenesis model for 4 simulation runs (only parameters that that vary over sets are presented)

6 DISCUSSION

Graphs are flexible and general data structures, but due to their combinational, orderless nature graph comparison poses some intrinsic problems that cannot be simply omitted during the development of new graph matching algorithms. First and foremost, graphs are computationally cumbersome. Direct comparison of two graphs requires enumeration of all sub-substructures and tackling with elements order. The exponential cost of such procedure makes construction of efficient graph metric infeasible. The more practical approach uses graph topological features as a set of comparison criteria. The graph descriptors can be selected according to their computational complexity and ability to retain information about graph structure. For instance, we can take into account densities of graphs, method of graph generation, distribution of degrees or connectivity, etc. Different criteria may result in different conclusions, therefore it is difficult to design graph matching method which would be suitable for all cases. Hence, the selection of graph descriptors used to compare graphs should be performed with respect to domain-specific knowledge.

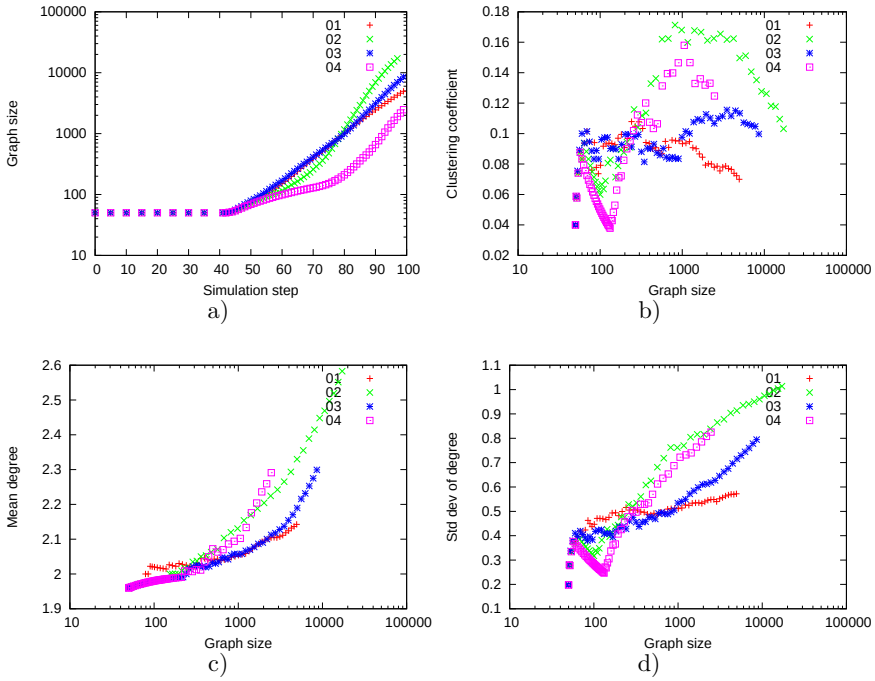


Fig. 11. The evolution of graph topology measures during angiogenesis simulation: a) simulation step dependence of network size (logarithmic Y scale), b) graph size dependence of clustering coefficient (logarithmic X scale), c) graph size dependence of mean degree (logarithmic X scale), d) graph size dependence of standard deviation of degree (logarithmic X scale)

Moreover, not the size of graph but distribution of its edges should be the main discriminating factor. Values of useful graph descriptor should mainly depend on the set of edges, not being dominated by the number of vertices.

Descriptors implemented in *Graph Investigator* can be grouped in terms of dependence on the graph size or considering the level of their generality. Using first criteria we can distinguish descriptors computed as statistical moments of certain vertex attributes which are independent of graph size, and can be used to compare networks with significantly different number of vertices. The examples of such descriptors are clustering coefficient (Equation (2)) and betweenness centrality (Equation (9)). On the other hand, descriptors such as Platt index (Equation (7)) or Wiener index (Equation (4)) depend on graph size (even after normalization), and generally are intended for comparison of similar-sized graphs. The scalar descriptors presented in Section 3.1 are more specific, in the sense that they capture well-defined properties of the graph. The algebraic descriptors, especially multidimensional ones, store many different topological features of graph but not in a straightforward way.

The graph descriptors can be aggregated in the form of feature vectors and used to embed graphs into metric space, where graph similarity/dissimilarity measures can be computed easily e.g. as Euclidean distance.

The biological context of the presented examples does not constrain our program to these types of applications. The rich set of implemented descriptors can be used in any other field of research that generates networks.

7 CONCLUSIONS

Graph Investigator provides a variety of network analytic tools ranging from feature generation to visualization. It enables to obtain distinctive insights into network structure by employing numerous descriptors from graph theory. The program was built with a special focus on biological networks as this kind of data grows constantly and its analysis has a wide range of applications. We reported three typical use cases: comparison of brain vascular network with artificial networks, analysis of vertices importance in cortical connectivity network of macaque monkey and inspecting how parameters of angiogenesis model affect structure of generated vascular networks. We believe that graph-theoretical approach to biological data analysis may bring many benefits, therefore developing software aimed at network exploration is a task of great importance.

Graph Investigator was designed to be flexible and easily extended. Graph descriptors can be added as *plugins*. The use of *Java* programming language provides its portability and makes the development simpler and faster. Currently our application is at beta stage, therefore all suggestions and contributions to its development are welcome.

In the nearest future we plan to extend the set of available graph descriptors and implement feature selection methods that allow for evaluation of different metrics using selected criteria. We also consider adding graph kernel functions as robust tool for non-linear analysis.

Acknowledgments

This work is financed by the Polish Ministry of Science and Higher Education, Projects No. 3 T11F 010 30 and No. N N519 579338. The authors thank Dr. P. Topa for providing angiogenesis simulation data and valuable comments.

Appendix

1 AVAILABLE DESCRIPTORS

Here we present the rest of network descriptors available in *Graph Investigator*. The first table contains scalar descriptors assigned to graph. Next, we list vertex

and edge descriptors intended for local topology analysis. Statistical moments of these metrics can be used to generate general graph descriptors. *Graph Investigator* enables to compute mean, standard deviation, skewness and kurtosis of vertex or edge descriptors.

1.1 General Graph Descriptors

Descriptor	Remarks
Randić Connectivity Index [51]	$\chi(G) = \sum_{(v,w)\in E}(k_v k_w)^{-1/2}, \quad \lvert E\rvert\chi(G) \leqslant \frac{\lvert E\rvert}{\lvert V\rvert-1}$ Connectivity measure derived from chemical graph theory
General Connectivity Index [63]	${}^{l}\chi(G) = \sum_{P_l\subset G}\left(\prod_{w\in V_P} k_w\right)^{-1/2}$, where P_l denotes path of length l, and V_P all vertices that belong to this path
Zagreb Index M_1 [52]	$M_1(G) = \sum_{v\in V}(k_v)^2$
Zagreb Index M_2 [52]	$M_2(G) = \sum_{e\in E} w_e$, where w_e is the weight of edge e. For unweighted graphs $\forall_{e\in E}\ w_e = 1$
Modified Zagreb Index ${}^{m}M_1$ [64]	${}^{m}M_1(G) = \sum_{v\in V}(k_v)^{-2}$
Modified Zagreb Index ${}^{m}M_2$ [64]	${}^{m}M_2(G) = \sum_{e\in E}(w_e)^{-1}$
Total Adjacency Index	$A(G) = \sum_{v\in V} k_v = 2\lvert E\rvert$
Modified Total Adjacency Index	${}^{m}A(G) = \sum_{v\in V}(k_v)^{-1}$
B Index [53]	$B(G) = \sum_{v\in V}\frac{k_w}{d_v}$, where d_v is the vertex distance (see 1.2) for vertex v
Density of edges	$Den(G) = \frac{2m}{n(n-1)}$, where $m = \lvert E\rvert$
Information of vertex degrees	$I_{vd} = \sum_{v\in V}(k_v \cdot \log_2 k_v)$ Reflects connectivity and topological complexity in terms of number of branches, cycles, cliques etc.
Radius of graph	$r(G) = \min_{v\in V} e_v$, where e_v is eccentricity (see 1.2) of vertex v
Graph diameter	$diam(G) = \max_{u,v\in V} d(u,v)$, Reflects density of graph connections, achieving its maximal value for paths, and minimal for cliques.
Total Walk Count [26]	Counts all paths of all lengths in the graph and depends on the size, cyclicity and branching of the graph, quantifying property called *labyrinthicity*.
Efficiency [2]	$E(G) = \frac{1}{n(n-1)}\sum_{u,v\in V, u\neq v}\frac{1}{d(u,v)}$, Measures the traffic capacity of a network and reflects its parallel-type transfer ability.
Heat Content, Heat Content Coefficients, Heat Kernel Trace, Heat Kernel Zeta Function [49]	Parameterized descriptors based on heat kernel matrix.

1.2 Vertex Descriptors

Descriptor	Remarks
Vertex distance	Sum of distances between v and all other vertices from graph G, $d_v = \sum_{w\in V} d(v,w)$
Eccentricity	Maximum distance between vertex v and any of the remaining graph vertices, $e_v = \max_{w\in V} d(v,w)$

Vertex B Index	$b_v = \frac{k_v}{d_v}$, where d_v is vertex distance for vertex v
Randić Shortest Path Index	${}^m\chi(G) = \sum_{P \in \mathcal{P}_m} \left(\prod_{u \in V(P)} k_u \right)^{-1/2}$, where $\mathcal{P}_m = \{P(v,w) \subset G : d(P(v,w)) = m\}$
Page Rank [66]	Vertex importance measure based on graph random walks model. Determines probability of turning up in vertex v after long-time random walk
Hubs and Authorities measure [67]	Evaluates authority of a vertex on the basis of link structure
Local efficiency [2]	$E_{loc}(v) = E(G_v)$, where G_v is a subgraph of neighbors of v and $E(\ldots)$ is graph efficiency.
Closeness	Mean distance to each other vertex.

1.3 Edge Descriptors

Descriptor	Remarks
Edge connectivity	$EConn(e) = k_v \cdot k_w$
Range of edge [68]	$g(e) = d_{G'}(v,w)$, where $G' = (V, E \backslash \{e\})$
Edge frequency [69]	The edge frequency for edge e, is a number of shortest paths, which contain edge e
Edge betweenness	Measures relative importance of an edge in shortest-path transfer through graph edges.

REFERENCES

[1] Newman, M. E. J.—Watts, D. J.—Strogatz, S. H.: Random Graph Models of Social Networks. Proceedings of the National Academy of Sciences of the United States of America, Vol. 99, 2002, pp. 2566–2572.

[2] Boccaletti, S.—Latora, V.—Moreno, Y.—Chavez, M.—Hwang, D. U.: Complex Networks: Structure and Dynamics. Physics Reports, Vol. 424, 2006, pp. 175–308.

[3] Képès, F. K.: Biological Networks. World Scientific Pub. Co., Inc. 2007.

[4] Podani, J.—Oltvai, Z. N.—Jeong, H.—Tombor, B.—Barabási, A. L.—Szathmáry, E.: Comparable System-Level Organization of Archaea and Eukaryotes. Nature Genetics, Vol. 29, 2001, pp. 54–56.

[5] Arodź, T.: Clustering Organisms Using Metabolic Networks. Computational Science-ICCS 2008, pp. 527–534.

[6] Arodź, T.: Conservation of Edge Essentiality Profiles in Metabolic Networks Across Species. Complex Sciences, pages Complex Sciences, Springer 2009.

[7] Csermely, P.—Ágoston, V.—Pongor, S.: The Effciency of Multi-Target Drugs: The Network Approach Might Help Drug Design. Trends in Pharmacological Sciences, Vol. 26, 2005, pp. 178–182.

[8] Yildirim, M. A.—Goh, K. I.—Cusick, M. E.—Barabasi, A. L.—Vidal, M.: Drug-Target Network. Nat. Biotechnol, pp. 1119–1126, 2007.

[9] Rubinov, M.—Knock, S. A.—Stam, C. J.—Micheloyannis, S.—Harris, A. W. F.—Williams, L. M.—Breakspear, M.: Small-World Properties of

Nonlinear Brain Activity in Schizophrenia. Human brain mapping, Vol. 30, 2009, pp. 403–416.

[10] Pachou, E.—Vourkas, M.—Simos, P.—Smit, D.—Stam, C. J.—Tsirka, V.—Micheloyannis, S.: Working Memory in Schizophrenia: An EEG Study Using Power Spectrum and Coherence Analysis to Estimate Cortical Activation and Network Behavior. Brain Topogr. 2008.

[11] Ponten, S. C.—Bartolomei, F.—Stam, C. J.: Small-World Networks and Epilepsy: Graph Theoretical Analysis of Intracerebrally Recorded Mesial Temporal Lobe Seizures. Clinical Neurophysiology, Vol. 118, 2007, pp. 918–927.

[12] Supekar, K.—Menon, V.—Rubin, D.—Musen, M.—Greicius, M. D.: Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimer's Disease. PLoS Computational Biology, Vol. 4, 2008.

[13] `https://networkx.lanl.gov/wiki`.

[14] Batagelj, V.—Mrvar, A.: Analysis and Visualization of Large Networks. Graph Drawing Software, 2003.

[15] Huisman, M.—Van Duijn, M. A. R.: StOCNET: Software for the Statistical Analysis of Social Networks. Connections, Vol. 25, 2003, pp. 7–26.

[16] Brandes, U.—Wagner, D.: Analysis and Visualization of Social Networks, Graph Drawing.

[17] Forman, J. J.—Clemons, P. A.—Schreiber, S. L.—Haggarty, S. J.: SpectralNET – An Application for Spectral Graph Analysis and Visualization. BMC Bioinformatics, Vol. 6, 2005, p. 260.

[18] `http://www.netminer.com`.

[19] `http://www.orgnet.com`.

[20] `http://jung.sourceforge.net`.

[21] `http://prefuse.org`.

[22] Fouard, C.—Cassot, E.—Malandain, G.—Mazel, C.—Prohaska, S.—Asselot, D.—Westerhoff, M.—Marc-Vergnes, J. P.: Skeletonization by Blocks for Large 3D Datasets: Application to Brain Microcirculation. IEEE Int. Symposium on Biomedical Imaging Macro to Nano, pp. 89–92, 2004.

[23] Kaiser, M.—Hilgetag, C. C.: Nonoptimal Component Placement, but Short Processing Paths, Due to Long-Distance Projections in Neural Systems. PloS Computational Biology, Vol. 2, 2006, pp. 805–815.

[24] Kötter, R.: Online Retrieval, Processing, and Visualization of Primate Connectivity Data from the CoCoMac Database. Neuroinformatics, Vol. 2, 2004, pp. 127–144.

[25] Choe, Y.—McCormick, B. H.—Koh, W.: Network Connectivity Analysis on the Temporally Augmented C. Elegans Web: A Pilot Study. Neuroscience Abstracts, Vol. 30, 2004.

[26] Rücker, G.—Rücker, C.: Walk Counts, Labyrinthicity, and Complexity of Acyclic and Cyclic Graphs and Molecules. Journal of Chemical Information and Computer Science, Vol. 40, 2000, pp. 99–106.

[27] Watts, D. J.—Strogatz, S. H.: Collective Dynamics of Small-World Networks. Nature, Vol. 393, 1998, pp. 440–442.

[28] Kaiser, M.: Mean Clustering Coefficients: The Role of Isolated Nodes and Leafs on Clustering Measures for Small-World Networks. New Journal of Physics, Vol. 10, 2008.

[29] Wiener, H.: Structural Determination of Paraffin Boiling Points. Journal of the American Chemical Society, Vol. 69, 1947, pp. 17–20.

[30] Platt, J. R.: Influence of Neighbor Bonds on Additive Bond Properties in Paraffins. The Journal of Chemical Physics, Vol. 15, 1947.

[31] Gordon, M.—Scantlebury, G. R.: Non-Random Polycondensation: Statistical Theory of the Substitution Effect. Transactions of the Faraday Society, Vol. 60, 1964, pp. 604–621.

[32] Todeschini, R.—Consonni, V.: Handbook of Molecular Descriptors. Wiley-VCH, New York 2000.

[33] Freeman, L. C.: A Set of Measures of Centrality Based on Betweenness. Sociometry, Vol. 40, 1977, pp. 35–41.

[34] Brandes, U.—Pich, C.: Centrality Estimation in Large Networks. International Journal of Bifurcation and Chaos in Applied Sciences and Engineering, Vol. 17, 2007.

[35] Newman, M. E. J.: A Measure of Betweenness Centrality Based on Random Walks. Social Networks, Vol. 27, 2005, pp. 39–54.

[36] Fan, R.—Chung, K.: Spectral Graph Theory. Conference Board of the Mathematical Sciences CBMS, No. 92, 1994.

[37] Farkas, I. J.—Derényi, I.—Barabási, A. L.—Vicsek, T.: Spectra of Real-World Graphs: Beyond the Semicircle Law. Physical Review E, Vol. 64, 2001.

[38] Godsil, C. D.—Royle, G.: Algebraic Graph Theory. Springer 2001.

[39] Tutte, W. T.: How to Draw a Graph. Proc. London Math. Soc., Vol. 13, 1963, pp. 743–768.

[40] Bagrow, J. P.—Bollt, E. M.—Skufca, J. D.: Portraits of Complex Networks. EPL (Europhysics Letters), Vol. 81, 2008, p. 68004.

[41] Roweis, S. T.—Saul, L. K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science, Vol. 290, 2000, p. 2323.

[42] Tenenbaum, J. B.—Silva, V.—Langford, J. C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, Vol. 290, 2000, p. 2319.

[43] Mika, S.—Schölkopf, B.—Smola, A. J.—Müller, K. R.—Scholz, M.—Rätsch, G.: Kernel PCA and De-Noising in Feature Spaces. Advances in neural information processing systems, Vol. 11, 1999, pp. 536–542.

[44] Wang, H.—Chen, S.—Hu, Z.—Zheng, W.: Locality-Preserved Maximum Information Projection. IEEE Transactions on Neural Networks, Vol. 19, 2008, pp. 571–585.

[45] Barabási, A. L.—Albert, R.: Emergence of Scaling in Random Networks. Science, Vol. 286, 1999, p. 509.

[46] Eppstein, D.—Wang, J.: A Steady State Model for Graph Power Laws. Arxiv preprint cs. DM/0204001, 2002.

[47] `http://www.amiravis.com`.

[48] Fruchterman, T. M. J.—Reingold, E. M.: Graph Drawing by Force-Directed Placement. Software – Practice and Experience, Vol. 21, 1991, pp. 1129–1164.

[49] Xiao, B.—Hancock, E. R.—Wilson, R. C.: Graph Characteristics from the Heat Kernel Trace. Pattern Recognition, Vol. 42, 2009, pp. 2589–2606.

[50] Xiao, B.—Hancock, E. R.—Wilson, R. C.: A Generative Model for Graph Matching and Embedding. Computer Vision and Image Understanding, Vol. 113, 2009, pp. 777–789.

[51] Randic, M.: Characterization of Molecular Branching. Journal of the American Chemical Society, Vol. 97, 1975, pp. 6609–6615.

[52] Gutman, I.—Trinajstic, N.: Graph Theory and Molecular Orbitals. Total-Electron Energy of Alternant Hydrocarbons. Chemical Physics Letters, Vol. 17, 1972, pp. 535–538.

[53] Bonchev, D.—Buck, G. A.: Complexity in Chemistry, Biology, and Ecology; Chapter: Quantitative Measures of Network Complexity. Springer 2004.

[54] Erdös, P.—Rényi, A.: On Random Graphs. Publicationes Mathematicae Debrecen, Vol. 6, 1959.

[55] Foggia, P.—Sansone, C.—Vento, M.: A Database of Graphs for Isomorphism and Subgraph Isomorphism Benchmarking. Proccedings of the 3rd IAPR TC-15 International Workshop on Graph-based Representations, 2001.

[56] `http://134.95.56.239/home.asp`.

[57] `http://www.biological-networks.org/?pageid=25`.

[58] Bassett, D. S.—Bullmore, E.: Small-World Brain Networks. The Neuroscientist, Vol. 12, 2006, p. 512.

[59] van Essen, D. C.: Organization of Visual Areas in Macaque and Human Cerebral Cortex. The Visual Neurosciences, Vol. 1, 2004, pp. 507–521.

[60] Uka, T.—de Angelis, G. C.: Contribution of Middle Temporal Area to Coarse Depth Discrimination: Comparison of Neuronal and Psychophysical Sensitivity. Journal of Neuroscience, Vol. 23, 2003.

[61] Duhamel, J. R.—Colby, C. L.—Goldberg, M. E.: Ventral Intraparietal Area of the Macaque: Congruent Visual and Somatic Response Properties. Journal of Neurophysiology, Vol. 79, 1998, pp. 126–136.

[62] Duffy, C. J.—Wurtz, R. H.: Medial Superior Temporal Area Neurons Respond to Speed Patterns in Optic Flow. Journal of Neuroscience, Vol. 17, 1997, pp. 2839–2851.

[63] Kier, L. B.—Hall, L. H.: Molecular Connectivity in Structure-Activity Analysis. John Wiley & Sons Inc. 1986.

[64] Nikolic, S.—Kovacevic, G.—Milicevic, A.—Trinajstic, N.: The Zagreb Indices 30 Years After. Croatica chemica acta, Vol. 76, 2003, pp. 113–124.

[65] Bonchev, D.—Rouvray, D. H.: Complexity in Chemistry, Biology, and Ecology; Chapter: Quantitative Measures of Networks Complexity. Springer 2004.

[66] Gori, M.—Maggini, M.—Sarti, L.: Exact and Approximate Graph Matching Using Random Walks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27,2005, pp. 1100–1111.

[67] Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, Vol. 46, 1999, pp. 604–632.

[68] Watts, D. J.: Small Worlds: The Dynamics of Networks Between Order and Randomness. Princeton University Press, Princeton, NJ, USA 1999.

[69] Girvan, M.—Newman, M. F. J.: Community Structure in Social and Biological Networks. Proceedings of the National Academy of Sciences, Vol. 99, 2002, p. 7821.

[70] Topa, P.: Towards a Two-Scale Cellular Automata Model of Tumor-Induced Angiogenesis. Lecture Notes in Computer Science, Vol. 4173, 2006, p. 337.

[71] Topa, P.: Dynamically Reorganising Vascular Networks Modelled Using Cellular Automata Approach. Cellular automata – 8th international conference on Cellular automata for research and industry, Lecture Notes in Computer Science, 2008, pp. 494–499.

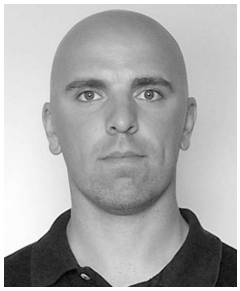
[72] Carmeliet, R. K.—Jain et al.: Angiogenesis in Cancer and Other Diseases. Nature, London 2000, pp. 249–257.

[73] Dzwinel, W.—Blasiak, J.: Method of Particles in Visual Clustering of Multi-Dimensional and Large Data Sets. Future Generation Computers Systems, Vol. 15, 1999, pp. 365–379.

Wojciech Czech received the M. Sc. degree in 2007 (Computer Science) from the AGH University of Science and Technology in Krakow, Poland. He is currently a Ph. D. student in Computer Science at the same university. His experience includes five years of work as software developer in telecommunications and science. His research interests concentrate on structural pattern recognition, graph comparison algorithms, complex networks and feature selection algorithms.

Witold Dzwinel is Full Professor in Computer Science and he has been Deputy Head in Department of Computer Science at AGH University of Science and Technology in Krakow since 2002 to the present. Between 1988 and 1992 he held senior researcher position in the Joint Institute of Nuclear Research and then between 1999 and 2003 in Minnesota Supercomputing Institute (MSI) at the University of Minnesota. His research concentrates on high performance computing in modeling of complex phenomena in physics, bio-sciences and in pattern recognition methods and algorithms. The method of particles as a universal heuristic in modeling of complex systems is his main point of interest. He is the author and co-author of about 160 papers in computer science and computational physics. He teaches courses in Pattern Recognition Methods, Modeling of Complex Systems and Computer Architectures.

Slawomir Goryczka received the M. Sc. degrees in 2006 (Computer Science) and 2007 (Mathematics); both from the AGH University of Science and Technology, Poland. His experience includes two years of work as an Operations Research Analyst in the airline market. He is currently a Computer Science Ph.D. student at Emory University in Atlanta, GA. His research interests include different aspects of complex networks and their dynamics, data privacy and security, distributed environments, operations research optimization techniques and random graphs.



Tomasz Arodz is an Assistant Professor at the Department of Computer Science at Virginia Commonwealth University in Richmond, Virginia, USA. He holds M. Sc. (2003) and Ph. D. (2007) degrees in computer science from AGH University of Science and Technology in Krakow, Poland. He also has an M. Sc. (2009) degree in biotechnology from the Jagiellonian University, Krakow. His research interests include bioinformatic analysis and inference in complex biological networks, statistical and computational analysis of evolution of proteins and the protein universe, and pattern recognition and machine learning for inference in biomedicine.

Arkadiusz Zbigniew Dudek is an Associate Professor in the Hematology, Oncology and Transplantation (HOT) Division, Department of Medicine of the University of Minnesota. He received his M. D. in 1989 from the Medical University of Warsaw, Poland. He received internal medicine residency training in Harrisburg, Pennsylvania and subspecialty training in Hematology, Oncology and Transplantation at University of Minnesota. He received his Ph.D. in oncology from Military Medical Academy of Warsaw in 1998. His clinical interests include phase I and II clinical trials in solid tumors, malignant melanoma, kidney cancer, and thoracic malignancies. His laboratory research is focused on cancer angiogenesis, tumor vaccines and preclinical novel drug testing.