Chapter 6
On the Trail of Nature: Collecting Scientific Evidence

Over every mountain there is a path, although it may not be seen from the valley.

— Theodore Roethke

To get closer to understand the nature of paths we need two kinds of data about the same networked system like the Internet, or the Bridges of Königsberg. First, we need at least an approximate network connecting its nodes and a large number of paths collected from real traces of packets. Using the words of the Bridges of Königsberg problem, we need the network representation of the lands and bridges (Fig. 3.2) and the footprints of people's afternoon walks.

In the last two decades, the flurry of network science [2] in all fields (biology, physics, sociology, technology) has resulted in the reconstruction of thousands of networks lying behind real-world systems [9]. Ranging from the classic Kevin Bacon game[1] over the network of Holywood movie actors, through the metabolic and social networks to the sexual contact of people, we now have systematically collected well-organized and publicly available data repositories about real-world networks (e.g., SNAP [17]). So downloading and computing something interesting over the network representation of cell metabolism in our cells is now an afternoon of laboratory work for an undergraduate student. What about the paths? Well, gathering paths seems to be a very different task compared to inferring simple connections in a network. The techniques working for the identification of network edges are generally not usable for gathering paths. Recall the Bridges of Königsberg again as an illustration. The map of downtown Königsberg is easy to get. Just jump into a map store and buy one, or draw an approximate map after a few days of walking in the streets of the city. The map or the network of the city is a form of public information. But what about the paths? Well, the paths belong to people. The paths describe the habits of people and tell us about them. About their favorite

[1] https://oracleofbacon.org/.

A. Gulyás et al., *Paths*, https://doi.org/10.1007/978-3-030-47545-1_6

places, the location of their homes and even about their health (if they prefer long or short walks). The nature of paths seems to be somewhat confidential. Some people may talk about it and give their names, others may talk about it anonymously and others may ignore you if you ask them about their paths.

Although gathering information about paths is not particularly easy, it is not hopeless either. Now we present four very different systems for which both the network data and the path data can be obtained to an appropriate extent. Our collection here will be based on the recent study of Attila Csoma and his colleagues about paths [6].

6.1 Flight Paths

When you board a plane of some airlines and get seated, you can find many things stuffed into the rear pocket of the seat in front of you. There are life-saving instructions, maps of the aircraft with the locations of exits, a sanitary bag, but there are also airline's magazines. In these magazines, among the advertisements about the most attractive flight destinations, there are usually nice maps showing all the flights operated by your airlines. If we could collect all magazines from the back pockets of all airlines, then we could easily reconstruct the flight network of the world, by considering the airports as the nodes of our network and the flights between them as the edges, no matter which airlines operate them. Although it would be quite time-consuming, it is absolutely doable.

Fortunately, there is a much simpler method for constructing the flight map of the world. Since flight information is public, there are public online data repositories which accumulate all the information about the flights all over the world. For example, the OpenFlights [21] project collects such data and makes the whole database publicly accessible. By listing, for example, all the flights of US airlines, the flight map of the US can be drawn (see Fig. 6.1).

Therefore, the reconstruction of the flight network is not rocket science, having the online datasets at hand. What about the paths? Well, a path is the multi-flight travel of somebody between the departing and destination airports through the flight network. Having a path means that we know the detailed flight information including the flight transfers for a given passenger or a set of passengers. Path information reveals how people choose transferring options at various airports, in cases where there is a lack of direct flights between the source and destination. Knowing a large number of paths is equal to knowing all the transfers of passengers for their trips, which is not something we should know without their consent and as such, there are no online databases for them. How can we obtain paths then? Well, we can take an "indirect" path to these paths. There are various flight-trip-planner portals that offer tickets between arbitrary sources and destinations all over the world. On such websites we can plan our whole journey and buy the tickets online. We can safely assume that many of the passengers buy their tickets using similar websites. So, what we can do is observe flights between randomly picked

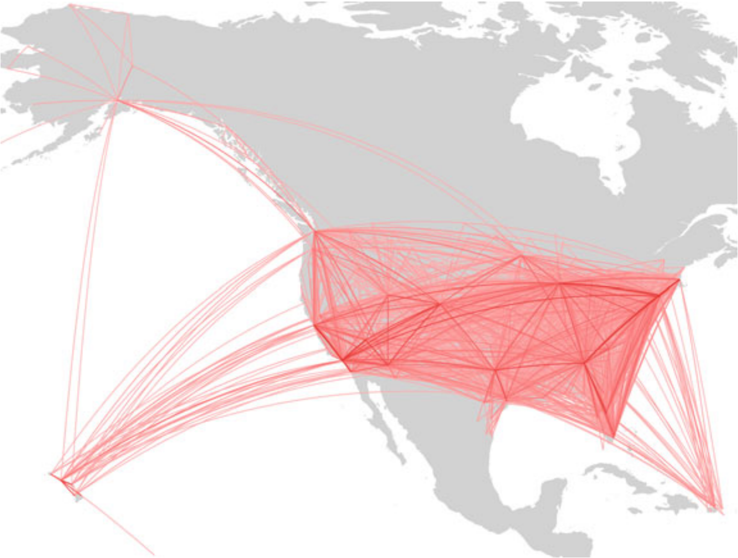

Fig. 6.1 The flight network of the US

airports and consider these offerings as paths that passengers could really choose for their trips. Gathering thousands of such flight offerings can give us a fairly usable approximation of paths used by real people, without raising confidentiality issues. Note that a particular trip can be the result of intricate business interactions among many different airlines and the passenger itself, so similarly to the Internet, the airport network also seems to be working without central coordination. What expectations may we have about the paths coming out of such a network? For starters, consider the position of Hungary's flagship airlines (which has stopped flying recently) in the flight system of the whole world.

MALÉV Hungarian Airlines was the principal airline of Hungary from 1946 to 2012. It had its head office in Budapest, with its main operations at Budapest Liszt Ferenc International Airport. In its best years, Malév operated direct flights between Budapest and New York, and Budapest and Moscow. In this respect, Moscow and New York could be connected by a path of length two through Budapest. What can we say about this two-step path offered by Malév? Well since one could travel from Moscow to Budapest and from Budapest to New York with Malév, we must say that this path is usable by passengers. However, Moscow and New York are huge metropolises with airports serving around 30 and 60 million people respectively, while Budapest airport is used by around 10 million people in a year. So, connecting these cities through the relatively small airport of Budapest looks a bit odd and we may suspect that the great majority of people would use other airports (e.g., Heathrow, Charles de Gaulle, Schiphol or Frankfurt) for changing flights between Moscow and New York. Although the background and the way of operation of the flight network are very different, we can suspect a similar underlying hierarchy of the airports as we have seen in the case of the military or the Internet.

6.2 Paths from a Word Maze

Word games are fun and entertain people regardless of their age. The Last and First game, for example, is frequently played between children and parents or grandparents. The essence of the game is to say a word which begins with the final letter of the previous word. For example, the word chain camel → lion → napkin → nest → tiger → raven can be the result of an afternoon game between granny and grandchild. Wait a minute! Doesn't this look like a path? A path of words? Sure it is! But instead of leading to somewhere, the aim of this chain is to go on as long as granny is awake and the grandchild is not bored. In this respect, the game does not have a destination (at least in terms of words). But can we twist the game a little bit so that the word paths will lead somewhere? Word ladder games are designed just for this purpose.

In a word ladder game, players navigate between fixed length source and destination words step-by-step by changing only a single letter at a time. For example, the word path fit-fat-cat is a good solution of a game with source word "fit" and target word "cat". This path is now very similar to our flight paths, in the sense that they have a definite source and destination and "transfers" can be made between words. Is there a public repository accumulating solutions of word ladder games played by people? Well, luckily there is[15]. Recently, Attila Csoma and his colleagues have developed a word ladder game for smartphones in a framework of a scientific project, and collect the word paths of people. After the users install the game, they are asked to transform a randomly picked three-letter English source word into an also random three-letter target word through meaningful intermediate three-letter English words by changing only a single letter at a time. The word paths entered by the users are collected anonymously. Fortunately, word path game solutions do not seem to be as confidential as flight information, as hundreds of users shared thousands of word paths (despite the clear deficiencies of a game developed by university researchers). These paths can be considered as the footprints of humans navigation over the word morph network of the English language.

More specifically, the collected paths are footprints of the process by which people master their navigational skills in the network lying behind the game. The word morph network is a network of three-letter English words, in which two words are connected by an edge if they differ in only a single letter at the same position (see Fig. 6.2). For example, the word "FIT" is connected to the word "FAT" as they differ only in their middle letter. "FAT" is linked to "CAT" as they differ in their first letter, but "FIT" and "CAT" are not connected in this network since they differ in more than one letter. The paths collected from players are paths in this network and reflect valuable information about how people try to navigate between nodes.

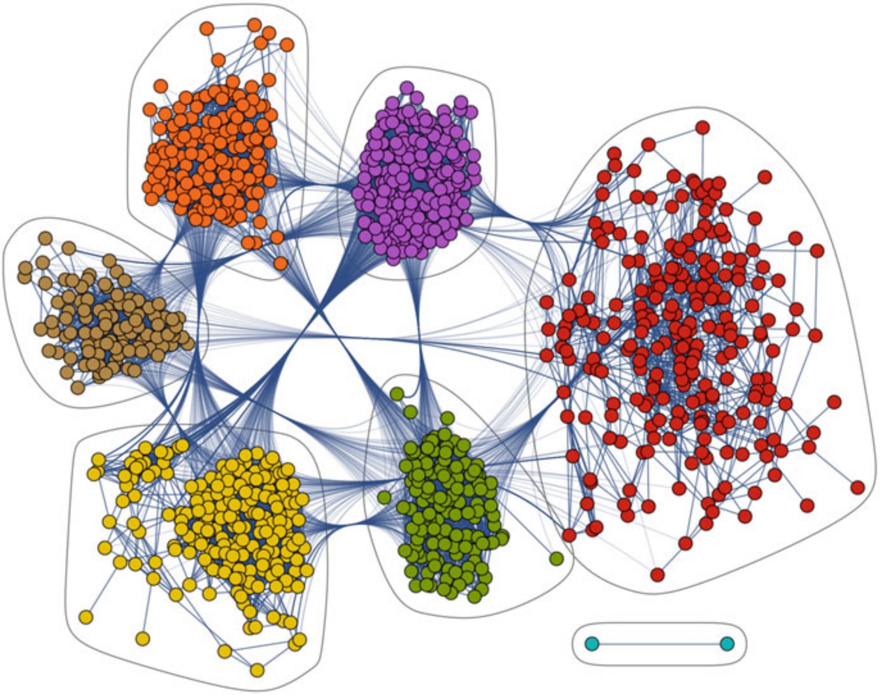

Fig. 6.2 The word morph network is a network of three-letter English words, in which two words are connected by a link if they differ only in a single letter. For example, "FIT" is linked to "FAT" as they differ only in the middle letter, but "FIT" and "CAT" are not neighbors in this network since more than one letter differs in them

Figure 6.3 shows a small portion of the word morph network and illustrates two solutions for a puzzle between source and target words "YOB" and "WAY".

What can we expect from these word paths? How will they look? Will we find "odd" paths and "regular" paths similarly to the chain of commands in the military and flight paths? As a sanity check, we present a common finding the players reached after playing some games. They realized that words are not equal in this game and some words can be used for various functions. The most basic puzzles, like the "FIT" → "CAT" one, can be solved by simply getting closer and closer to the destination in terms of matching letters. In "FIT" there is one matching letter with "CAT", in "FAT" there are two matching letters, while in the destination "CAT" all letters are matched. How about the "TIP" → "ALE" puzzle? This is much more complicated since the consonants and vowels are at completely opposite positions. In this case, the above strategy simply doesn't work. Now the players have to find words with back-to-back consonants or vowels, where such letters can be swapped. For example, the "TIP" → "TIT" → "AIT" → "ALT" → "ALE" is a solution, where the intermediate word "TIT" is just there to turn to "AIT" at which vowels are back-to-back at the front, which then can be changed to "ALT" at which consonants are

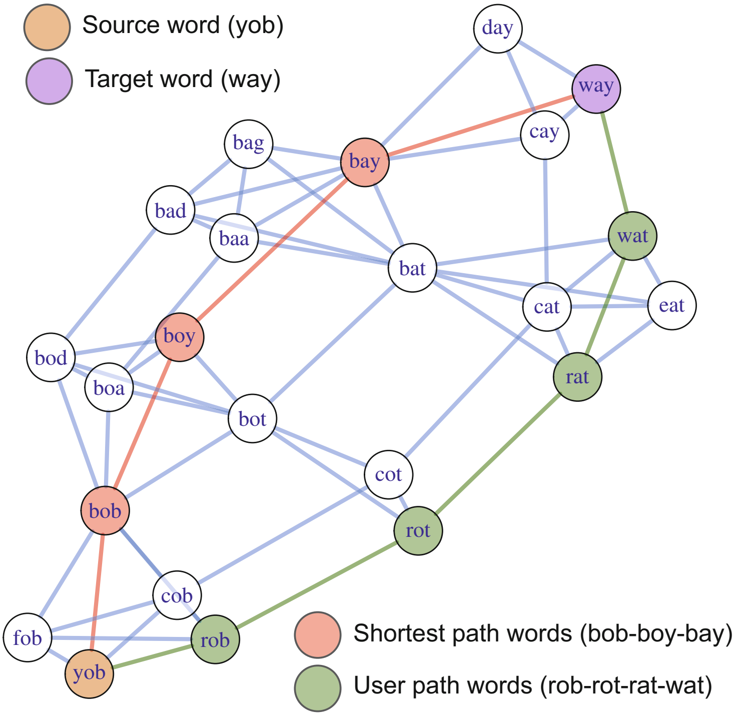

Fig. 6.3 A word morph game example with source and target words "YOB" and "WAY". A shortest solution is displayed in red, while a solution given by a specific player is shown in green

back-to-back at the end and is just one step from "ALE". People quickly memorize such "trade" words like "AIT" or "ALT" and reuse them in further puzzles. So, it seems that there is also some underlying logic in this simple word game. But is this logic similar to what we can find on the Internet or in the flight network?

6.3 Internet Paths

The reconstruction of the network to which the Internet has evolved after more than three decades, grasped the attention of many researchers worldwide. As the Internet is built over electronic devices, its topology could be reconstructed by collecting all the connection-related data residing in each of its constituting nodes (i.e., computers). However, unlike the airport network where the flights between airports is public information, the connection information between Internet providers is not easy to obtain. The traffic agreements between internet providers are usually kept confidential. Therefore, it seems that we cannot even get the underlying network of the Internet in a straightforward way, not to mention the paths we are curious about. It turns out, however, that some Internet hacks can help us find the paths. In this respect, the Internet is a unique platform for researching paths.

To get a picture of how packets go through the Internet, we have to understand some fundamentals of computer networking first. A packet traveling between computers is nothing more than a few bits of information encoded as electronic signals. Every packet has a source and destination address and a payload which should be delivered to the destination. Packets usually do not change and do not think, which is in high contrast with the people walking through the bridges of Königsberg. Instead of the packets, the computers (i.e., the lands) "think". What does a computer (let's refer to them as nodes in a network context) do when receiving a packet? First, it looks into the destination address. If the destination address is the current node, then it "consumes" the packet. After extracting the payload data, the packet is destroyed. If the destination address is not the node receiving the packet, it has to find out how to forward the packet to its destination. Who tells the node how to find this out? People! Not ordinary people of course, but networking people whose job it is to operate networks. There is a routing table in every node, which is very similar to road signs. It indicates the next turn a packet should take on the way to a specific region of the network.

Consider Fig. 6.4 as an example. There are seven computers marked with letters (A, B, C, D, E, F, G) forming a very simple network of seven edges. Now suppose that D wants to send a packet to G. As a first step, it creates a packet containing the source address D, the destination address G and also the payload data, just like a postal letter. Now node D has to send the packet to G. The situation of D is extremely simple as it has no choice where to send the packet, its only option is B. Quite the contrary, at node B (as it is not the destination) there are plenty of options to forward to. How will B decide? Well, a capable network operator configured B to solve such situations. The operator creates the routing table of B, from which B can read the next step of the packet destined to G: it must be given to C. Similarly, C is instructed to send packets with destination address G, to node G. As a result finally G receives the packet successfully. The full operation can be made according to the routing tables in the nodes (see Table 6.1). From these routing tables, all the paths

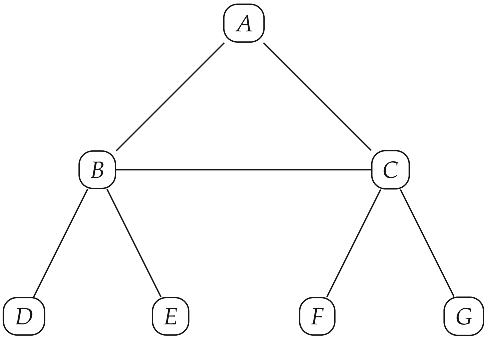

Fig. 6.4 A simple network of computers

Table 6.1 Possible setting of routing tables for the network in Fig. 6.4

Routing tables of the nodes		
Node	Destination	Next step
B	D	D
	E	E
	A	A
	C, F, G	C
C	F	F
	G	G
	A	A
	B, D, E	B
A	B, D, E	B
	C, F, G	C
D	ALL	B
E	ALL	B
F	ALL	C
G	ALL	C

between any pairs of nodes can be reconstructed. The problem is that the nodes of the Internet belong to various networking corporations, which do not intend to disclose the routing tables. So, collecting the paths in such a way is not an option. What can we do then to get our paths?

Fortunately, there is something in computer networks which all networking companies are scared of. So scared, that they implement several mechanisms to detect and avoid them. These daemons are called loops. Consider that in our simple computer network in Fig. 6.4, every node is administrated by a distinct company, i.e., a different operating person. Consider that the operator of A notices that its direct connection to C is weak, e.g., it provides a slow connection. So the operator of A sets the routing table in A to forward every packet destined for C, F and G to B, avoiding the laggy direct connection to C. Independently, B also considers its connection to C as pretty weak and forwards all packets heading to C, F and G to A (see all the modified routing tables in Table 6.2, with the modifications shown in boldface).

Now, what happens with the packet destined for G after these tiny, uncoordinated modifications in the routing tables? Well, it starts at D as seen before. B sends it to A according to its routing table, but A sends it back to B, B sends it back to A, A sends it back to B ... and so on forever. There is an infinite loop between B and A. After some time, node B and A are only occupied by looping the packet infinitely, which eats their resources, pointlessly generates a lot of heat, and most importantly ruins the operation of the whole network. You may think that the routing settings are carefully negotiated between networking operators, so such things could not happen. Well, routing settings are usually well negotiated, but the human factor is always there. Misconfigurations happen every day on the Internet. One of the most famous examples was in 2008 when due to a routing table misconfiguration in a

Table 6.2 Setting of routing tables leading to a loop for the network in Fig. 6.4

Routing tables of the nodes		
Node	Destination	Next step
B	D	D
	E	E
	A	A
	C, F, G	$\boldsymbol{A}$
C	F	F
	G	G
	A	A
	B, D, E	B
A	B, D, E	B
	C, F, G	$\boldsymbol{B}$
D	ALL	B
E	ALL	B
F	ALL	C
G	ALL	C

node of the Pakistan Telecom, a large portion of YouTube's traffic was hijacked and discarded in Pakistan.

Thus, loops are dangerous things in computer networking and one should immediately detect and avoid them. The current solution for that is to include so-called time-to-live (TTL) information in the packets. This is a simple number which is decremented by every node the packet visits. If this number becomes zero, the packet will be destroyed even if it hasn't reached its destination. This way, bad configurations will have limited effects as packets cannot travel for an infinite time between the nodes. When a packet's TTL becomes zero and it is not at its destination, that is a good sign that something is wrong with the network. In this case, the node which destroys the packet sends an alert to the source address found in the packet stating that something is wrong. And this is where the approximate tracing of packets becomes possible.

Consider the following hack. We want to know the nodes on the path towards a destination node D from source node S. First, we send out a packet from S and set its TTL value to 1. This packet will reach one of S's neighbors (let's say A), which will destroy the packet and notify S that something went wrong. From this notification, we record, that our packet has visited node A. Now we start again and send out the packet, but this time setting its TTL to 2. The packet would not be destroyed by node A as its TTL becomes 1 when A decrements it. So A will forward the packet to somewhere, let's say to B. Since at B the TTL is decremented again, it becomes zero, so B destroys the packet and sends back a notification to S that something went wrong. At S, we record that the packet has visited B so the path to B is $S \to A \to B$. The process continues until a large enough TTL setting lets our packet reach its destination. Sounds a bit complicated but this is all we have. This method (called traceroute) gives us approximate paths and we can use this method from any node connected to the Internet, even from your laptop. Fortunately, there are public datasets which contain such Internet paths collected from thousands of different locations. These datasets (see for example the website of the Center for Applied Internet Data Analysis[4]) can give us millions of paths from which an approximate map of the Internet can be recovered.

How can we construct the topology of the network from paths? Suppose that we have three paths: *Path 1*: A → B → C, *Path 2*: A → B → D → E, *Path 3*: E → C → F → A. By analyzing *Path 1*, we see that there are nodes A, B and C and there are edges between A and B, and between B and C. From this, we can draw a network shown in Fig. 6.5.

Now we analyze *Path 2*. We realize, that there are also nodes D and E in the network, and we locate two edges: B → D and D → E (see Fig. 6.6). Finally, the observation of *Path 3* adds a new node F and three edges: E → C, C → F, and F → A. So, after processing the three paths, we get the network in Fig. 6.7. After processing more and more paths, we will have more and more appropriate pictures of the whole network.

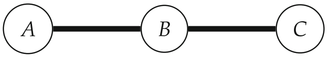

Fig. 6.5 Constructing a network based on its paths, Phase 1

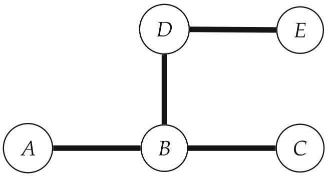

Fig. 6.6 Phase 2

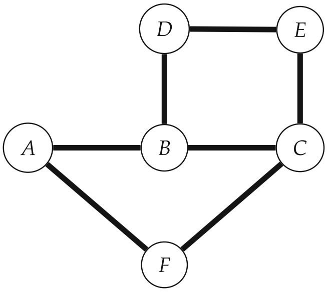

Fig. 6.7 Phase 3

6.4 Paths from the Human Brain

The human brain is one of the most complex networks one could imagine. Understanding even parts of its functionality is extremely challenging and is still one of the biggest mysteries of human life. Here we are interested in the paths inside the brain: the paths over which information can travel between different parts of the brain. Getting realistic paths from inside the human brain is extremely hard, if not impossible. As a consequence, almost all current studies concerning path-related analysis simply assume that signaling uses shortest paths, meaning that we suppose brain signals follow the shortest possible path in the brain. Similarly to these studies, we have to accept that we cannot get paths out from the brain in a direct manner. In the case of the Internet and the flight network, the confidentiality of the path related data was the main obstacle of getting direct paths. In the case of the brain, we just simply don't have the appropriate technology (yet) which could identify the paths for us. What can we do then? Is there a similar hack for the brain that we used over the Internet? What kind of data is currently available about the flow of information inside the brain? We will go through these questions in the following paragraphs.

The Human Genome Project (Fig. 6.8) was one of the biggest endeavors of mankind and was surrounded by the most remarkable scientific collaboration across many nations. Its target was to determine the sequence of nucleotide base pairs that make up the human DNA. Upon its completion, at a press conference at the White House on the 26th June 2000, Bill Clinton evaluated the resulting map of the human genome as: "Without a doubt, this is the most important, most wondrous map ever produced by humankind."

A similar endeavor started in 2011 when the Human Connectome Project was awarded by the National Institutes of Health. This project is targeted to construct the "map of the brain", i.e., to discover the structural and functional neural connections within the human brain. The structural map means that we locate specific brain areas (these will give us the nodes) and the physical connections (which will give us the edges) between them. How can one do this without slicing up somebody's brain? Well, this is what the "non-invasive" brain mapping methods are used for. With a quite complicated method called DSI (Diffusion Spectrum Imaging), the diffusion of water molecules can be observed inside the brain. To get a picture of how DSI works, think about constructing a road network by observing only the movement of cars at various observation points throughout the area you want to map. You cannot see the roads themselves, but you can see the cars at these observation points and you can write the direction and intensity of their movements. By collecting all this information from the observation points, after some non-trivial computerized post-processing, we can create an approximate map of roads and cities in the given area. Interestingly, the process is very similar to the operation of WAZE, a popular navigation software (now owned by Google), where the positions of WAZE users are collected in an anonymized database. In this case, however, the exact map is drawn by volunteer editors, using the draft map deduced from the database. In DSI, the cars

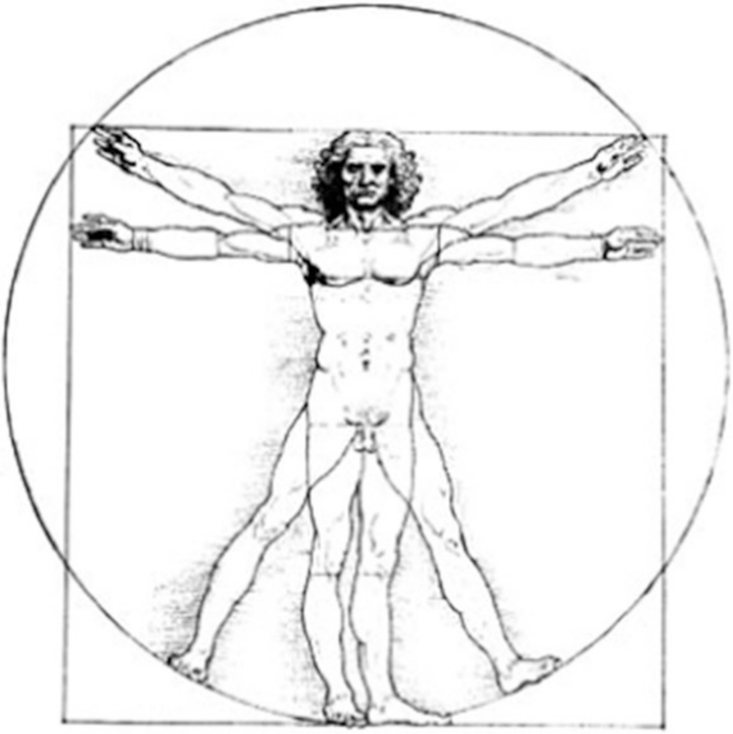

Fig. 6.8 The Vitruvian Man depicting normal human body proportions is often used to symbolize The Human Genom Project as Leonardo da Vinci created it in 1490, exactly a half a millennium before the project began in 1990. [Public Domain; Leonardo da Vinci via Wikimedia Commons]

are water molecules, which are observed at various points in the brain by using MRI (Magnetic Resonance Imaging) devices. A picture about the human connectome, i.e., an approximate picture of one's map of neural connections in the brain, obtained via DSI, can be seen in Fig. 6.9.

Thanks to DSI we can have one's connectome, i.e., we have the network over which our paths form. What can we say about the paths? It seems that at this time we can say nothing about them in a direct manner. But there is something we can do to at least estimate brain paths better than simple shortest paths? fMRI[20] (functional Magnetic Resonance Imaging) is a method with which one can reason about brain activity. With fMRI, the blood oxygenation of various regions in the brain can be measured. Since blood flow and oxygenation are correlated with brain activity (active brain regions use more energy and require a higher level of oxygen in the blood), the changes in blood oxygenation reveal the neural activity. Back to our city-roads-cars analogy, fMRI is quite similar to the task of reasoning about the operation of a city, by observing the density of cars in its various districts.

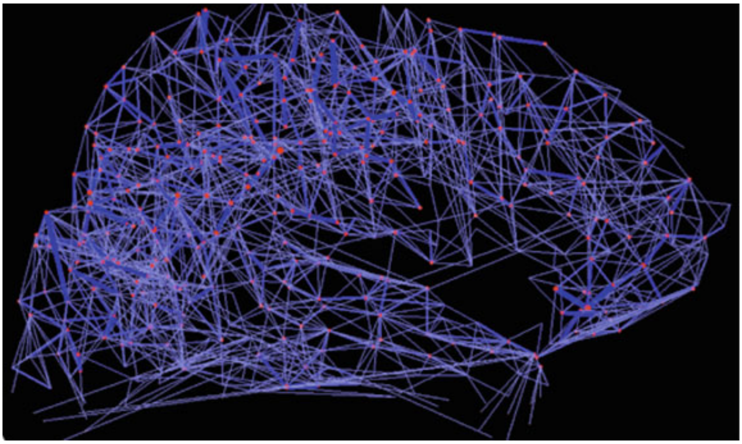

Fig. 6.9 The human neural network in the brain reconstructed via DSI, from Patric Hagmann et al. “Mapping the structural core of human cerebral cortex”. In: PLoS biology 6.7 (2008), e159

How can we approximate paths in the brain? Well, DSI delivers an approximate “network” of the brain, meaning that it gives us the nodes and the physical connections (the bridges in the Königsberg analogy) between them. The fMRI gives a different “network” in which brain regions are not physically, but functionally or logically connected, meaning that they frequently act together, so they seem to implement similar functionality. Can we make use of some trick and infer something path-like from these data? Here is what we can do. By combining structural (DSI) and functional (fMRI) data, we estimate paths through which neural signals might propagate using the following hack. First, we have to identify the sources (i.e., the starting node) and destinations (the nodes where the path ends) of our paths. From the fMRI signals, we can identify brain regions, which frequently exhibit neural activity at the same time. Simultaneous activity hints that these brain regions are working on the same task and are likely to exchange information in the form of neural signals. We identify these simultaneously active brain regions as the source-destination pairs of our paths. Now we have to figure out the path between these sources and destinations. In cases where there is a lack of information, we could determine the shortest path between the endpoints of our paths using, for example, Dijkstra’s algorithm over the structural connectivity network obtained from DSI. Figure 6.10 shows an illustrative brain network of 15 nodes. Over this network, we would like to approximate the possible signaling path between regions 1 and 15. The shortest path approximation will give the $1 \to 5 \to 12 \to 15$ path for this. In fact, most studies in the related literature use this simple approximation.

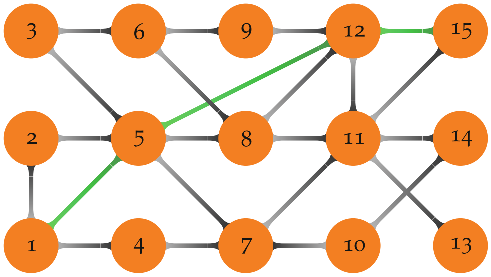

Fig. 6.10 Inferring path from the human brain using the shortest path assumption

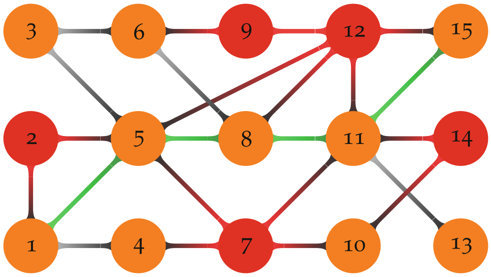

Fig. 6.11 Shortest path over the active subnetwork at a given time instant

Due to the extreme complexity of the brain, as of now, we do not have direct information about the paths inside, but we can do slightly better than simple shortest paths. We can use the fMRI to identify regions with neural activity and from the DSI network, we can exclude the inactive regions during signal transmission between the endpoints of our paths. We can do this because inactive regions are not likely to pass on any information. By excluding inactive regions, we will get the active subnetwork for every information exchange we are curious about. Figure 6.11 shows the same network we can see in Fig. 6.10, but the red regions (2,7,9,12,14) are inactive, and thus are excluded from the path approximation. Therefore, we will find the shortest path between 1 and 15, but we cannot step onto the red regions. The shortest path in this new scenario is 1 → 5 → 8 → 11 → 15, which is longer than the shortest path in the original DSI network.

While we cannot validate with empirical data whether these paths (see Fig. 6.12) are actually used for the flow of neural signals, we can at least consider these paths as lower bounds on the length of the real brain paths.

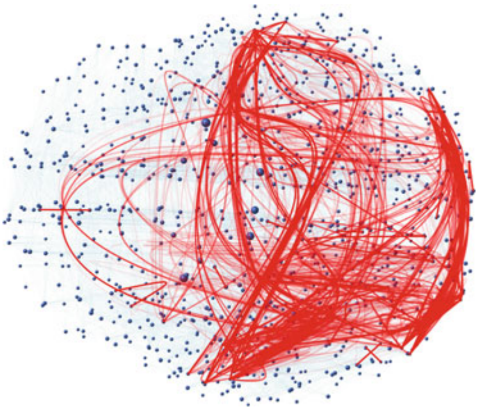

Fig. 6.12 Empirical paths in the human brain