***Chapter 4: following data***

*4.0 Introduction*

Telling stories about how people move around inside the physical spaces of laboratories and sequencing centers reveals only some of the topology of bioinformatics. Twenty-first century science involves a further set of spaces. In bioinformatics, the relationships between spaces are not only physical, but are also influenced by other kinds of proximities: Ethernet cables, microwave links, and shared computer networks form a second kind of ‘space’ that must be mapped and taken into account. This chapter will pay close attention to these ‘virtual’ spaces – how they are arranged, who can access them, and how people interact within them.[[1]](#footnote-1) The landscape of scientific communication and the possibilities for collaboration, interaction, and exchange are fundamentally reworked by electronic networks; these spaces must be considered autonomous means through which scientific actors and objects might be ordered, distributed, and brought into proximity with one another. In other words, they present new modes through which scientific knowledge can be produced and certified.

As one might expect of a digital discipline, much of the important space and motion in bioinformatics is virtual: shared disk drives, local networks, servers farms, compute farms, and the World Wide Web comprise another, less immediately apparent, landscape in which bioinformatics is enacted. In the labs I visited, these virtual spaces were not only harder to find and harder to access, but more intricate and more difficult to navigate than the physical spaces. Indeed, a large part of learning to do bioinformatics is the process of learning to find one’s way around the appropriate virtual spaces – having the expertise to know *where* some piece of information or code might be was often as valuable and as valued as knowing what to do with it. This chapter draws on my experiences at the at the Broad Institute and the European Bioinformatics Institute (EBI) in order to describe what these virtual spaces of bioinformatics look like, how people move around in them, and how this motion constitutes the *doing* of bioinformatics.

The first part of this chapter details how biology becomes virtual in the first place – how material samples become data through the sequencing ‘pipeline.’ This process is at the center of bioinformatics: showing how the material becomes the virtual is to show how biology becomes digital. What becomes apparent in the description of the ‘pipeline,’ however, is that this flattening into data is a complex and contested process – samples do not just automatically render themselves into data. Although a pipeline suggests linearity, what we find here is that is it ‘out of sequence’ – it is a messy, contingent process. Following the data to its origins shows how its generation depends on care and judgment. Much work is done to make standardized, virtual objects that are amenable to computers and networks. In the second part, I examine another way in which biology is rendered into data. Producing ‘ontologies’ means creating standardized, common languages for speaking about biology. The uniformity and universalism of biology is manufactured during bioinformatic work. Such standard, computable objects are necessary for bioinformatic work to be possible. Bioinformatics standardizes and flattens biological language into data. Third, I examine how bioinformatics objects are moved around in virtual space. The rendering of biology into standard, computable formats, allows knowledge to be produced by the careful movement and arrangement of data in virtual space.

What emerges here is a new picture of biological work and biological knowledge. Biological work consists in the rendering of biological stuff into data and the proper arrangement of this data in virtual space. Neither the creation of data nor its movements in virtual space are wholly linear, fluid, or frictionless processes. Data must be carefully formed into its digital shape and its arrangement must accord with standardized structures such as standardized data formats, ontologies, and databases. These structures set constraints on what biological objects are and how they can be talked about.

*4.1 Pipelines*

At the center of bioinformatics is a process that transforms the real into the virtual – renders biological samples into data. The ‘pipeline’ is a spatial metaphor that recurs often in bioinformatics. As such, it is also an appropriate metaphor to use for describing the transition from old to new forms of biological work. The pipeline moves us from the material to the virtual and from the lab bench to the computer network.

‘Pipeline’ is a word used to describe the series of processes applied to an object in order to render it into some appropriate final form. Pipelines can be either physical (involving transmission and transformation of actual objects) or virtual (involving transmissions and transformation of data); but they are often both, and most often describe the processes through which actual objects (DNA from an organism) is rendered into virtual form (DNA in a database). The pipeline is the method through which the actual becomes the virtual, through a carefully ordered set of motions through physical and virtual space.

When we think of a pipeline we might immediately think of an oil pipeline or a water pipeline in which a liquid is transported directly along its length. Readers familiar with computers might also think of the ubiquitous ‘pipe’ command in Unix (represented by ‘|’) in which one program or command takes as its input the output from the previous program; by ‘piping’ several programs together, the user creates a space through which data can flow. In both these cases, there are two key concepts. First, pipelines are directional: the water or the data must flow down the pipe, following the single path laid out for it only and never moving side-to-side or going backwards. Second, pipelines require liquidity: solid objects are generally inappropriate for pipes and piping programs together requires that inputs and outputs of adjacent programs in the pipe are designed to match one another.

In 2008, I spent several months observing the ‘sequencing pipeline’ in action at the Broad Institute. In their Sequencing Center almost all the work is based around the pipeline – it is the metaphorical object around which work is ordered. This is the set of methods and processes that are used to transform raw DNA into sequence data that is of an appropriate form for submission to an online database such as GenBank. Samples arrive at the Sequencing Center in many forms: cells growing in media, as microorganisms, blood, tissues, or even whole large organisms. In the first place it is the task of the Molecular Biology Production Group (MBPG) to extract the DNA and prepare it for sequencing. The MBPG’s role consists of three stages. First, the “DNA preparation” team subjects the incoming DNA samples to a number of tests to check its purity and quality. They then shear the DNA into random pieces of a size suitable for sequencing using a purpose-built machine that subjects the DNA to hydrodynamic forces. The DNA is then stored in a -20°C freezer in the MBPG lab area. Second, the “ligation team” is responsible for the 5-6 day long process of library production. A library of DNA is a large collection of short DNA fragments that together represent a complete segment of DNA (a whole chromosome, for example); since a library is constructed using many copies of the same segment, most of its parts will be represented in many of the small pieces. The DNA fragments are chemically treated so that they are incorporated into specially engineered, ring-shaped piece of DNA called plasmids. Completed libraries are also stored in freezers. Finally, the “transformation team” produces finished agar plates for handoff to the Core Sequencing Group. *E. coli* bacteria are mixed into the plasmid solution and by rapid heating or electric shock, induced to take up the plasmids. Workers must then spread the bacteria-containing solution thinly over a nine-by-nine inch agar plate infused with an antibiotic. The engineered plasmid includes an antibiotic resistance gene such that *E. coli* which have not taken up a plasmid will die, while those that have incorporated a plasmid will grow into colonies as they are incubated overnight. Each resulting colony on the agar plate should contain many copies of a particular DNA fragment in its plasmids.[[2]](#footnote-2)

Each of these tasks is delegated to a team consisting of four of five workers, usually with one acting as a coordinator. Every sample receives a barcode when it enters the pipeline. As a sample moves around the laboratory this barcode is repeatedly scanned and the results stored in a Laboratory Information Management System (LIMS). For example, a sample would be scanned when it is taken in or out of a freezer, or when it undergoes a particular process such as ligation. Through the LIMS it is possible to track the status and location of any sample in the Center at any time. By querying the LIMS, workers can discover what work is available for them to do, and where to find it. For instance, a member of the ligation team could find the freezer location of all the samples that had undergone DNA preparation and were ready for ligation.

In these early steps, much attention is paid to space and motion. The time and effort required to move samples through the lab is carefully monitored and analyzed. The location of equipment (centrifuges, freezers, picking machines, etc.) and lab benches is optimized so that workers can move the samples quickly and efficiently through the lab spaces. These processes are constantly scrutinized in order to maintain the highest levels of quality and efficiency. Incubation times, solution concentrations, and other details are monitored and modulated to increase yield.

Once the agar plates are incubated the responsibility of the MBPG ends and the Core Sequencing Group takes over. Their work can be divided into six phases. First, in the “picking” phases, a large robot designed by a company called Genetix uses a high-resolution camera to scan the agar plates to produce digital images. These images are processed by software which identifies colonies that are good candidates for ‘picking’ – that is, colonies that show good growth and are not too close to one another. A robot arm then moves a 96-pin head over the plate, picking the selected colonies and transferring each to an individual well of a 384-well tray or plate containing glycerol solution.[[3]](#footnote-3) The agar plates are then discarded and the 384-well plates are incubated for a further 15-18 hours. The samples are then once again placed in a freezer. Second, the “TempliPhi” process generates further amplification, using a PCR-like process to increase the amount of DNA about ten million-fold in four hours. Third, in the “sequencing” step, specially designed dyes are added to the samples. In order to get a sense of what is involved in this process I quote from a detailed description:

The sequencing team starts by transferring the DNA from the TempliPhi Eppendorf Twintec(TM) plate into two other Eppendorf Twintec(TM) plates and diluting the DNA mixture with water. The top of the red plate (the TempliPhi plate) is transferred into the top and bottom halves of one blue plate, so there are two copies of DNA in one blue plate. The bottom half of the red plate is transferred into another blue plate, and the red plate is then discarded. Once this is done, two types of dye are added to each blue plate; “forward” sequencing dye is added to the top half of the plate, and “reverse” sequencing dye is added to the bottom half of the plate. [[4]](#footnote-4)

The fourth and fifth steps are “ethanol precipitation” and “elution” respectively. These are both chemical processes designed to remove excess dye and other contaminants. These are far from automated steps. Each involves careful laboratory work, testing, and judgment calls. In picking, the agar plates are “visually checked” for approximate accuracy of counts and spacing; the 384-well plates are checked for a “hazy” appearance (indicating a growing colony) before sequencing; solutions used in the TempliPhi steps are checked each morning (randomly selected plates are weighed to ensure correct amounts of reagent are being added).[[5]](#footnote-5) This is not a straightforward sequence, but rather an interlocking set of checks and tests that require knowledge of the process chemistry.

It is in the sixth step, “detection” that the samples disappear wholly into the virtual. And it is in this step only that the samples are introduced into the sequencing machines. About 120 Applied Biosystems 3730 detectors, each costing several hundreds of thousands of dollars, sit together in a large single room – each is about six feet high and has a footprint of roughly three feet by two feet. Because of the cost of the machines and the speed at which they process samples, detection is both the rate-limiting and the cost-limiting step. As such the detectors are operated twenty-four hours per day, seven days per week. Inside the machines, the dyes attached in step three are excited by a laser and detected by a CCD camera. In a typical detector output, the four possible nucleotide bases are represented by four colored lines on a chart (red, green, blue, black).

These colored graphs are known as the “raw sequence traces.” They are stored by the sequencing machines, but they are not yet ‘finished’ sequence. As the raw sequence traces are generated by the detectors, two computational steps are automatically performed. The first is known as “base calling” and is usually done using software called “phred.” Phred analyzes the raw trace images and, based on the height and position of the peaks, makes a “call” as what the correct base is for each position along the sequence. In other words, it converts an image file to a text string consisting of the familiar As, Gs, Ts, and Cs. It also assigns a quality score to each base, reflecting its confidence in the call that it makes. The second automated step is “sequence assembly.” In the very first stages of the sequencing process, before the DNA was spliced into plasmids, the entire sequence was broken into random fragments. In order to generate the whole sequence, the individual reads (one from each well of the plates) must be reassembled. A program called “phrap” is used to do this. Phrap uses both the base calls and the quality scores from phred to determine the most likely assembly by searching for overlapping segments (it takes into account the fact that in regions where quality scores are low, overlaps could be due to random errors). Both phred and phrap are open source software packages developed in the early 1990s by Phil Green and his colleagues at Washington University, St. Louis.[[6]](#footnote-6)

Despite the sophistication of these algorithms, they are usually unable to match up every sequence – for almost every genome, gaps remain. It is the job of the “finishing” team to take over where phrap leaves off and to patch up the gaps. There are about forty finishers at the Broad – their work is highly specialized and requires detailed knowledge of both the biology of the organisms being sequenced, the intricacies of the sequencing process, and the assembly software. By querying the LIMS, a finisher can find samples in the queue that require finishing work. During my fieldwork at the Broad I spent several hours watching a highly experienced finisher at work. After retrieving the relevant sequences from the LIMS database, the finisher imports them into a graphical user tool that shows the overlapping sequence regions one above the other. Scrolling rapidly along the sequence on screen, the finisher made quick decisions about the appropriate choice of base at each position where there seemed to be a discrepancy: his experience allowed him to tell “just by eyeballing” where the phrap had made a mistake by placing TT in place of T, for instance. The graphical tool also allowed the finisher to pull up the raw traces – sometimes this was necessary in order to make a decision about which base call was the correct one. Where gaps existed, the finisher imported sequence data from sources that had not been used in the assembly (an online database for example) in order to begin to fill the hole. This work is often painstaking and relies crucially on the finisher’s judgment. This quotation describes the finisher’s reasoning process as he alters a single G to a C in the sequence:

Here now is a discrepancy in the consensus [sequence]. Usually it’s where we have a whole string of Gs like we have here; these reads here are fairly far away from where the insert is – I can tell just by their orientation – they’ve been running through the capillary [the key component of the detector] for a while when it hits one of these stretches... so it’s basically calling an extra G even though these high quality reads – you can tell by the color – are clearly correct, clearly calling a discrete number of Gs, and so again this one is a C.

In cases where it seems that there is insufficient data to fill a gap, the finisher may request further laboratory work on that sequence region – such requests are passed through the LIMS. This also requires careful consideration:

Finishers often find themselves in a catch-22: to select an appropriate laboratory procedure, they must understand the underlying sequence, but the sequence is missing. In practice, they must make educated guesses about the underlying sequence and the likelihood that various laboratory techniques will succeed. Their decision is influenced by the condition of the DNA near the gap; it is also influenced by the ability of their informatics tools to highlight those conditions. Most importantly, finishers’ decisions are guided by their skill and experience: whereas some experienced finishers may be able to close a gap based on the information already present in an assembly, less experienced finishers may feel they need laboratory work.

Under their contracts to the NHGRI and other funding bodies the Broad’s standard for ‘finished’ sequence requires no more than one error in every 10000 bases. A recent audit found error levels around one in every 250000 bases. Once a finisher is satisfied with the piece of sequence on which he or she is working, it is resubmitted to the LIMS. From here it must pass through a “final review” by a different finisher before it can be submitted to a GenBank as a ‘finished’ sequence.[[7]](#footnote-7) During this process the finisher runs a number of scripts on the sequence to check for quality and consistency and which aligns the piece of sequence to the entire genomic build. The submission process is also automated: a sequence is passed through a script which automatically generates the required metadata such as the coordinate systems, the sequence length, and the author names. This script also does a final check for gaps and writes standard features into a GenBank submission file. All progress, including the GenBank submission number, is recorded in the LIMS. Finished submission files are automatically uploaded to GenBank via FTP at 11pm each day where they are available to the world at 8am the following morning.

The integrity and fluidity of the sequencing pipeline is maintained by the LIMS. Without this computer tracking system, the various activities of the pipeline would remain disconnected from one another. It is the LIMS that makes it possible to think of a discrete-sample-thing flowing through the pipeline; it is only in the computer that the place-to-place motion of the samples is made sense of, or makes sense. This makes is problematic to think of the process as a clear-cut transition from actual sample to virtual sequence inside the detector; as soon as a sample is barcoded and entered into the LIMS it becomes a virtual as well as an actual object. The actual object in the lab would be meaningless without its virtual counterpart linking it to the pipeline. As the actual-virtual object progresses down the pipeline, however, it does become increasingly liquid – as it is further disconnected from its actual presence it is transformed into a more mobile form. Images such as figure 3.3 suggest the intricate labor that must be performed in the early stages of the pipeline as the sample is tracked in its minute movements around the lab. By the finishing stage however, the sample can be effortlessly pulled up onto a screen and, finally, dispatched instantly across hundreds of miles to the GenBank database in Bethesda, MD. The sequencing pipeline can be understood as a process of enabling DNA to travel.

But the pipeline does more than make things flow. The metaphor also serves to obscure the transformational and judgmental aspects of the sequencing process. Pipelines merely transport a uniform substance from one place to another. A sequencing *pipeline* (rather than a sequencing production line, for instance) suggests that the extracted text was already inside the body of the sequenced subject, waiting to be piped out (like oil or natural gas); it suggests that the As, Gs, Ts, and Cs are merely transported from the organismic body to hard-drives in Bethesda. The pipeline allows biologists to understand sequencing as ‘travel through space,’ rather than as an active process of extraction and construction shot through with difficult manual tasks and active judgment calls. Those directly involved in the day-to-day work of the pipeline rarely experience it as a linear flow. Figure 4.1 shows a process flow diagram depicting a far more intricate set of actions and interconnections that is suggested by a pipe. Organizational and work-flow charts on the walls of the Broad depicted something more like a dense network than a pipeline. The metaphor, however, serves to conveniently collapse the sequencing process for those who use sequence data; imagining the sequencing process as a linear flow serves to de-problematize the actual-to-virtual transformation and allows the sense that sequencing is an automatic, black-boxable activity. Such an elision is crucial in much of bioinformatics for which ‘the sequence’ is the central object. Understanding sequencing as a directed, unproblematic process – a simple linear motion through space – allows the output products (the digital renderings of the sequences themselves) to plausibly be used as proxies for the input materials (wet biological samples). Almost all of bioinformatics relies on this fact that sequences can be relied upon as straightforward extractions of biological material into biological data. This detailed description of the ‘pipeline’ shows how the production of biological knowledge (in this case sequence) depends crucially on carefully contrived motion through space. Such motion is far from automatic or linear – it is shot through with judgment calls and contingent processes. The production of ‘fluid’ and ‘universal’ bioinformatic objects depends on a highly situated process through which this ‘fluidity’ and ‘universality’ is constructed; bioinformation can travel and flow only because the motion through the pipeline serves to obscure its solidity and situatedness.

**[figure 4.1 about here]**

The pipeline is not intended to completely capture the features of a biological specimen. Indeed, such a task would be impossible. Rather, the material-virtual transition reduces the sample to a digital trace, flattens it into a text that can be computed, communicated, and manipulated in virtual space. Sequence data, consisting of a base call and a quality score, is already a structure to which biological sequences must conform. Following the data to its place of production shows how it is constructed for a particular purpose; it is crafted as a data-object ready to flow through computers and networks.

*4.2 Standards and ontologies*

Simply producing sequence data and putting it in GenBank would not be useful for biological research unless bioinformaticians and biologists already agreed about standards and formats for writing, storing, and reading such data. GenBank is are doing far more than just storing biological data – it is enacting common standards for the reading, writing, and storage of biological information. This section will examine how bioinformaticians go about developing and enforcing common languages. But this is not merely a problem of communication – it is also a problem of knowledge. By flattening language into data, it becomes possible to create ways of agreeing about objects and their proper descriptions.

For DNA sequences themselves, a standard shorthand was already in place: A, G, T, C, and U were in use before large-scale DNA sequencing began. As DNA sequencing became ubiquitous, the genetic code was supplemented by other letters such as R (G or A), B (G, T, or C) or N (A, G, T, or C). Such tolerance for ambiguity reflected the fact that sequencing was still sufficiently expensive that some information was better than none. Such a code is already a significant abstraction from a ‘physical’ DNA molecule, which might contain non-standard nucleotides, epigenetic markers, three-dimensional conformations, or other chemical variations.[[8]](#footnote-8)

Despite the standard AGCT, variation in coding was still problematic for bioinformatics. Different software programs, for example, required sequences to be submitted in different formats. By the late 1980s, however, the superior speed of Bill Pearson and Walter Lipman’s FASTA program, meant that its input format gradually became a standard for the encoding of sequences. Even though FASTA was superseeded by BLAST, FASTA format persisted.[[9]](#footnote-9) The format consists of a first line that begins with a ‘>’ on which information identifying the sequence is listed. The sequence itself is then listed on subsequent lines. The appearance of another ‘>’ in the file indicates the beginning of a new sequence. Although FASTA has been criticized because the identification line lacks a detailed structure, it remains the de facto standard for sharing and transferring sequence data. Its simplicity makes it particularly attractive for programmers who want to be able to parse sequence data into their programs quickly and easily. It is a simple but powerful data structure to which sequences must conform.

The ready existence of a widely agreed-upon code has made the sharing of sequence data relatively straightforward. The sharing of other kinds of biological data, however, has required more elaborate schemes. In particular, it is what is known as ‘annotation data’ that has caused the greatest problems. Annotation data includes all the information attached to sequence that is used to describe it: its name, its function, its origin, publication information, the genes or coding regions it contains, exons and introns, transcription start sites, promoter regions, and so on. The problem is that such data can be stored in a variety of ways; different descriptions in natural language (for example: ‘Homo sapiens,’ ‘H. sapiens,’ ‘homo sapiens,’ ‘Homo\_sapiens’), different coordinate systems, and different definitions of features (for instance, how one defines and delimits a gene) inhibit compatibility and interoperability. There are two kinds of solutions to these problems, one of which I will call ‘centralized’ and the other ‘democratic.’

The democratic approach is known as a Distributed Annotation System (DAS). As it was described in a seminar at the EBI, DAS is like a Web 2.0 ‘mash-up,’ like using Google to create a map of local pubs by pulling in data from different sources. The idea is the data remains in various ‘native’ formats in a large number of geographically dispersed repositories; the DAS server, however, knows how to talk to each repository and extract data from it on the fly. The EBI’s main database, known as *EnsEMBL,* works largely through a DAS. When a feature is requested through the *EnsEMBL* website, the website creates a standard URL that tells the DAS which information to retrieve; the DAS then queries the remote database and sends back the relevant data in XML format, which can be interpreted and displayed by the *EnsEMBL* website. DAS does not aim to translate all the data into a common language or format – instead, it can display data from multiple sources side-by-side on a website. Not only does this side-step problems of translation, but also avoids (since data is requested in real time over the web) time consuming efforts to keep data up-to-date. The inventors of the DAS system were explicit with their intentions for what kinds of biological work they hoped DAS would promote:

DAS distributes data sources across the Internet improving scalability over monolithic systems. This distribution of data encourages a divide-and-conquer approach to annotation, where experts provide and maintain their own annotations. It also permits annotation providers to disagree about a particular region, encouraging informative dissension and dialogue. The separation of sequence and map information from annotation allows them to be stored and represented in a variety of database schema. A number of different database backend alternatives could arise. The use of links as a method of referencing back to the data provider’s web pages provides even greater power of expression and content control.[[10]](#footnote-10)

DAS is not only a technical solution but also a mode of political organization for biology. The creators of DAS imagine a democratic biology in which the task of data management is shared and knowledge is a product of debate and negotiation. DAS does not allow biological data to travel anywhere – local databases must be compatible with the XML standards that the DAS server uses. However, DAS attempts to be minimal in its approach, requiring a relatively small investment on the part of the local database. DAS represents a simply constructed network that is built using local materials and using local methods; it is highly heterogeneous but can include a large amount of data because it does not need to be rigorously maintained or policed.

Centralized solutions, on the other hand, aim to bring all the data into a common format at a single location – software is written to translate data from various formats and deposit and maintain it in a master database. These are the most ubiquitous and fastest growing set of solutions to the problems of sharing biological data and manifest a philosophy opposite to that of DAS. These centralized solutions, called ‘ontologies’ tend to be top-down and heavily managed. Ontology is the branch of philosophy that deals with what things exist in the world: it is “the science of what is, of the kinds, structures of objects, properties, events, processes, and relations in every area of reality... Ontology seeks to provide a definitive and exhaustive classification of entities in all spheres of being.”[[11]](#footnote-11) For computer scientists, predominantly those concerned with information management and artificial intelligence, an ontology has a slightly different meaning:

a dictionary of terms formulated in a canonical syntax and with commonly accepted definitions designed to yield a lexical or taxonomical framework for knowledge-representation which can be shared by different information systems communities. More ambitiously, an ontology is a formal theory within which not only definitions but also a supporting framework of axioms is included.[[12]](#footnote-12)

Why would computer scientists care about making such dictionaries? If computers are to be able to reason with information, the language used to communicate such information to the machines must be standardized. Several ambitious attempts have been made to build exhaustive vocabularies for large-scale business enterprises. In 1981, the firm Ontek began developing ‘white collar robots’ that would be able to reason in fields such as aerospace and defense.

A team of philosophers (including David W. Smith and Peter Simons) collaborated with software engineers in constructing the system PACIS (for Platform for the Automated Construction of Intelligent Systems), which is designed to implement a comprehensive theory of entities, ranging from the very concrete (aircraft, their structures, and the processes involved in designing and developing them) to the somewhat abstract (business processes and organizations, their structures, and the strategies involved in creating them) to the exceedingly abstract formal structures which bring all of these diverse components together. [[13]](#footnote-13)

This is an attempt to create a “theory of the world” that can be used by computers. In biology the problem is not only getting computers to work with biological data, but also getting biologists to work with each other. Biological ontologies are supposed to solve the ‘data silo’ problem by creating *controlled vocabularies* for the sharing of biological information.

In the late 1990s, as the number of completely sequenced genomes grew, several senior scientists in charge of managing the genome databases for these organisms began to realize the need for a shared language. In particular, they needed a way of talking about the functions of genes, the central objects of biological interest. Several efforts had been made to create functional classification systems, but these “were limited because they were not shared between organisms.”[[14]](#footnote-14) Suzanna Lewis, who was in charge of FlyBase (the genome database for *Drosophila* fruit fly) reported on some of her emails from 1998:

Our correspondence that spring contained many messages such as these: “I'm interested in defining a vocabulary that is used between the model organism databases. These databases must work together to produce a controlled vocabulary” (personal communication); and “It would be desirable if the whole genome community was using one role/process scheme. It seems to me that your list and the TIGR [The Institute for Genome Research] are similar enough that generation of a common list is conceivable (personal communication).[[15]](#footnote-15)

In July 1998, at the Intelligent Systems for Molecular Biology (ISMB) conference in Montreal, Michael Ashburner suggested a simple hierarchical controlled vocabulary for gene function. His paper, ‘On the representation of “gene function” in databases’ was not well received – most participants considered it naïve. Afterwards, however, in the hotel bar, Lewis (representing FlyBase) met with Steve Chervitz (representing the yeast database, Saccharomyces Genome Database) and Judith Blake (Mouse Genome Informatics) and agreed to a common scheme for describing the functions of genes.[[16]](#footnote-16) This collaboration became the Gene Ontology (GO) consortium.

**[figure 4.2 about here]**

Ashburner was already aware of the work in artificial intelligence and medicine on ontologies and his proposal included the suggestion that GO be compatible with, or at least translatable to, more generalized ontologies.[[17]](#footnote-17) He argued that GO had to be more sophisticated than a mere list of terms – “the advantage of a structured graphs over a flat key-word list is that you could have a representation of part and whole, it’s easy to maintain, it’s easy to use, and you have information built into the structure of the graph, whereas with flat key-words, there is only one thing you can do with that: sort it alphabetically.”[[18]](#footnote-18) A structured graph looks somewhat like a flow diagram in which terms are linked together by arrows. Figure 4.2 shows a small, simplified section of the GO structured graph. The arrows represent how the GO terms are logically linked to one another – in this case how a cell is comprised of various parts and subparts. ‘Cell’ has two parts, the ‘cytoplasm’ and the ‘nucleus.’ The nucleus has three subparts, the ‘nucleolus,’ the ‘nucleoplasm,’ and the ‘nuclear membrane.’ Since this structure is programmed into the GO itself, this enables a simple computer program to read off these relationships from the hierarchical structure of the graph itself. In other words, the GO encodes a structure for biological objects.[[19]](#footnote-19)

Since 1998, GO has evolved into three distinct hierarchies, one describing biological processes (what process a gene product is involved in), one describing molecular function (what molecular mechanisms are involved), and one describing cellular components (where in the cell the gene product acts). The explicit aim is the unification of biology through a shared language: “all biologists now acknowledge that there is likely to be a single limited universe of genes and proteins, many of which are conserved in most or all living cells. This recognition has fueled a grand unification of biology; the information about the shared genes and proteins contributes to our understanding of all the diverse organisms that share them.”[[20]](#footnote-20) GO is designed to be flexible, to dynamically adjust to changing ideas in biology, and to be responsive to the biologists who are using it.[[21]](#footnote-21) Nevertheless, GO requires a centralized group of ‘curators’ or ‘editors’ who have a broad knowledge of the overall structure, scope, completeness, and consistency requirements of the ontology and maintain ultimate control over its terms.[[22]](#footnote-22)

GO has certainly not solved all communication and consistency problems in biology. This is, in part, because it was only designed to provide a language for talking about gene function. The success of the GO has inspired a host of ontologies in other biological domains: cell types, descriptions of environment, experimental techniques, human diseases, anatomy, pharmacogenomics, imaging methods, pathways, and human phenotypes all have their own ontologies. The Open Biomedical Ontologies (OBO) Foundry attempts to bring order to this proliferation by setting up rules and standards for the creation of ontologies themselves with the ultimate aim being “a suite of orthogonal interoperable reference ontologies in the biomedical domain.”[[23]](#footnote-23) In particular, OBO has developed a ‘relationship types’ ontology that specifies an ontology for the logical connectors (‘is\_a’, ‘part\_of’, ‘has\_part’, ‘has\_agent’, etc.) used by ontologies.[[24]](#footnote-24) Such ontologies of ontologies, or meta-ontologies, provide a framework for what their creators hope will be an all-encompassing language for describing biology and medicine.

Proponents of ontologies believe that they are the only reliable way of promoting the free exchange of experimental data in science. To make this point, Barry Smith (one of the founders of the National Center for Biomedical Ontologies) contrasts ontologies to ‘folksonomies’ of the kind enabled by the photo-sharing website Flickr ([www.flickr.com](http://www.flickr.com/)).[[25]](#footnote-25) Photos in Flickr are can be “collaboratively categorized” using “freely chosen keywords” based on the tags that individuals apply to their photos.[[26]](#footnote-26) How is it possible to choose between the myriad of individual sets of classifications if there are no constraints? Smith examines four possibilities: deferring to the authority of a ‘terminology czar’, using the first that comes along, using the best, or using an ontology based on “reality, as revealed incrementally by experimental science.”[[27]](#footnote-27) An ‘instrumental’ ontology, based on a particular model of how biology works, will sooner or later be superseded or out-competed. A ‘realist ontology,’ Smith argues, is the only way to make ontologies work. Terms in an ontology should correspond to “what one actually sees in a lab, not what is convenient.”[[28]](#footnote-28) More elaborately:

Ontology, as conceived from the realist perspective, is not a software implementation or a controlled vocabulary. Rather, it is a theory of reality, a ‘science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality.’[[29]](#footnote-29)

Ontologies are not static lists but rather dynamic structures that evolve with scientific ideas. Building ontologies, he concludes, should be like building scientific theories. The vision for biological ontologies is that they will enforce agreements about what objects exist and the relationships between them – objects in the biological world must be fitted into the structure of the ontology.

Some social scientists have taken for granted the important role that ontologies should play in encouraging collaboration and data-sharing. Sabina Leonelli argues that “Bio-ontology consortia function as a much-needed interface between bottom-up regulations arising from scientific practice, and top-down regulations produced by governmental agencies. They achieve this by focusing on practical problems encountered by researchers who use bioinformatic tools such as databases.”[[30]](#footnote-30) She suggests that the centralization of power in consortia like that for GO actually promotes epistemic pluralism in biological research. My ethnographic experience, however, suggests that ontologies in general, and GO in particular, are not universally accepted tools amongst working biologists. This is best appreciated by noting the fact that GO in fact has several competitors; although GO is the most widely used, biologists will often run their data against multiple gene ontologies before being satisfied that the result is plausible.[[31]](#footnote-31) At a lab meeting one principle investigator warned his students and collaborators: “GO is useless, I always ignore it ... it’s way too general.” Expressing her distrust of GO, another bioinformatician reminded me that ultimately the system was just maintained by “a bunch of post-docs” who just sat in a room reading scientific papers and deciding what terms to include. When I raised the subject of ontologies amongst biologists they often reminded of the quip, sometimes attributed to Ashburner, that “biologists would rather share their toothbrush than share a gene name.” Such reactions suggest a profound discomfort amongst biologists with the centralization of responsibility and the structuring of knowledge that ontologies impose. Biologists often describe their discipline as ‘messy’ compared to sciences like physics or chemistry; what is interesting is found in the exceptions rather than the rules, and careers have been built on the ability to navigate this uniqueness. This suggests why the kind of the standardization that ontologies offer is not always well received: biologists consider the freedom and flexibility of their categories and their language to be an advantage for investigating and describing biological systems. Conforming to standards, even if they are collaboratively developed, and speaking in controlled vocabularies may not be in a biologists’ self-interest, at least not in the short term.

The examples of ‘centralizing’ and ‘democratizing’ regimes I have described here entail particular political and technical visions of biology. Each recognizes the need for maintaining a balance between flexibility (in order to promote scientific innovation) and structure (in order to allow data sharing). Where they crucially differ is in how their visions of how biological expertise should be distributed in space. Ontologies imagine a grand unification of biology powered by a reduction of language to a universal and machine-readable form – one of the aims of the ontologists is making GO and other bio-ontologies compatible with OWL (Web Ontology Language). Biological knowledge will be produced largely from the resources of the Semantic Web, guided by a few experts in central locations.[[32]](#footnote-32) The alternative, as exemplified here by DAS, is a wiki-biology – multiple visions and multiple languages are allowed to flourish and compete for attention and certification. Biological knowledge will be distributed and heavily reliant on local expertise.

Gene Ontology attempts to constrain the shape and form of bioinformatic objects – it tries to determine the kinds of things that can exist in digital biology. But it also polices the relationships between them – it has consequences for biological knowledge because it establishes structures and hierarchies through which biological things can relate to one another. As we will see with respect to databases in the next chapter, the structures of information technologies exert powerful forces on how biologists think about organisms. Simultaneously, the Gene Ontology also has consequences for the disciplinary structure of biology – it establishes hierarchies between different groups of biologists. Gene Ontology shows how (just as we have seen elsewhere) bioinformatics is a transformation of objects, knowledge, and the organization of biological work.

In the short term, biology will continue to rely on ontologies to promote data-sharing. My aim here has been less to criticize such ontologies and more to show how specific technical solutions determine specific structures within which biologists must talk and act. Not only the movement of data, but ultimately the authorization of bioinformatic knowledge, depends on the organization and hierarchies within the biological community. But DAS is an alternative techno-social solution to the problem of data sharing. It suggests that the structures imposed by ontologies are not necessary but contingent – they are built by practicing biologists. Indeed, much of the work of bioinformatics is in generating these ways and means of allowing data to move around frictionlessly.

Another important example of this type of standardization is the work of the Genomic Standards Consortium (GSC). Since 2005, the GSC has attempted to extend the reach of standard vocabularies to cover the provenance of DNA samples. For instance, their Minimum Information about a Genome Sequence (MIGS) creates standards for reporting the geographic location at which a sample was collected, the time, the habitat (including temperature, light, pressure, pH, etc.), health of the organism, sequencing method, assembly method, extraction methods, standard operating procedures, and a range of other factors (all with associated standard vocabularies).[[33]](#footnote-33) To actually produce the information to meet such a standard, biologists would need to actually follow specific procedures and methods for gathering and processing DNA samples. It would mean measuring temperature, pH, using a particular assembly method, creating standard operating procedures, and so on. The standardization of language enforces particular ways of working and doing.

In attempting to standardize language, GO is also flattening biology. It takes the multiplicity and complexity of biological language and renders it into a data structure. GO suggests how bioinformatics involves the standardization and data-ization of biological knowledge. But the structure that GO creates does not just effect computers; it also affects how biologists think about biological objects and what they do with them. A standardization of terms contributes to a standardization of biological practice and biological knowledge. Like other technologies of virtualization, ontologies make biology more compatible with computing by reducing it to standard forms that can be coded, digitized, and shared through electronic networks.

*4.3 Virtual spaces*

What happens to this data once it enters the virtual realm? How does it get manipulated there? This section tracks how data is moved around in the performance of bioinformatics knowledge-making. The flattening and virtualization of samples and language into data allows the organization of data in space to become a kind of knowledge production. The value of bioinformatic work consists of this careful spatial ordering according to the structures of databases, data structures, and ontologies. Formatting, managing, and curating data are ways of arranging it into knowledge.

What do the virtual spaces of bioinformatics look like? In the most literal sense they are text on a computer screen. Most bioinformatics begins with a ‘Unix shell’ – a command prompt at which the bioinformatician can enter commands in order to navigate around. In the first instance, one might only need to navigate around the files stored on one's own computer using such commands as “cd” (change directory) and “ls” (list, which provides a list of all the files and other directories in the current directory). On almost all computers, files are accessed by means of a hierarchical tree of directories, which might be navigated by traveling up and down the branches using the “cd” command. In keeping with the tree metaphor, the topmost directory is usually known as “root” (labeled “/” in Unix). However, almost anything useful in bioinformatics will require accessing files on other computers; these might be the computers of colleagues sitting next to you, they might be networked hard-drives sitting in the corner of the lab or the basement of the building, or they might be servers at some far-distant location. In many cases, the physical location of the machine being accessed makes little or no difference and is often unknown to the user. The most common way of connecting to another computer is to use “ssh” or a “secure shell” – this is a system of communication that allows you to log into another computer remotely using a username and password. For example, to log into a computer called 'tulip' at the EBI I might type: *ssh tulip.ebi.ac.uk*. Tulip will prompt me for a username and password. If I am authorized to access tulip, the command prompt will reappear – the screen and the prompt might look exactly the same as if I was using my own computer, and I can now navigate around the tulip machine using exactly the same commands.

This sounds straightforward enough. However, access to such spaces is highly regulated. At the EBI access to all physical spaces is controlled by RFID cards – when I arrived I was provided with an ID on the first day. Access to virtual spaces is limited by username and password combinations – by contrast, it took over a week to arrange my access to all the commonly used computers. Moreover, not all computers are directly accessible – sometimes a series of logins is required, ssh-ing from one's own computer to computer A and then from A to B. Some computers can only be accessed by programs (those usually used for intensive calculation), some are dedicated to particular kinds of data (databases, for instance), some are for everyday use, some for use by particular users or groups, some for long-term storage, some for back-up, and some for hosting publicly accessible websites. Figure 4.3 gives a sense of the variety of machines involved and the complicated way in which they can be connected to one another. Bioinformaticians use metaphors of space in talking about how they move around these extended networks: tunnels, firewalls, routers, shells, and transfers all suggest a space in which both the user and the data can move around. As I learned to how to login to various machines and find my way around the network my questions would invariably be answered with diagrams of boxes (representing computers) connected by lines or arrows (representing potential ssh connections between them). Although bioinformaticians interacted with such systems only textually, such movement relied on a latent mental image of how such machines were virtually linked together.[[34]](#footnote-34)

**[figure 4.3 about here]**

What are the contents of such virtual spaces? Virtual spaces are inhabited by four types of objects. First, they contain data, either in files or in databases. Second, they contain computer programs (software), which might, at any given time, actually be using or manipulating data. Third, they contain directories, which, along with the databases, are the organizing structure for both the data and the software. Fourth, they may contain bioinformatian-users moving around in the space and manipulating data, software, or the directory structure. The way in which these four elements interact with one another is both intricate and tightly constrained. For instance, beyond the necessity of logging in to any machine, certain kinds of data may only be manipulated by certain individuals (often, those who created it). Further, users may not store data just anywhere or execute software on any computer: data must be placed into particular directories or directory structures and programs must be run on dedicated machines. The “farm” system in place at the EBI illustrates the care which is taken to control and monitor movement through the virtual space. The “farm” or “compute farm” is a set of computers (549 in November 2008) used by the EBI for running programs requiring intensive calculation or data manipulation. For most users, it is not possible to login into the farm directly. Instead bioinformaticians log into a computer called ‘farm-login’ which is connected to the farm. In order to run a computationally expensive program, users must submit a ‘job’ using a system called LSF. This program allocates the job to one or more of the machines in the farm. According to the internal Wiki:

All users of the farm initially have equal priority. The queueing system uses a policy called “Fair-share” to ensure equal distribution of work. LSF will dynamically change your priority depending on how much time you have recently consumed on the farm. If you have been running lots of jobs, LSF will lower your priority to allow other users’ jobs to be dispatched in preference to yours. Conversely, if you have not used much farm time, LSF will increase your priority with respect to other users.[[35]](#footnote-35)

The farm is designed to ‘farm out’ the computational work in an efficient and equitable manner. The metaphor of working the land is appropriate – if the total network of computers are imagined as a space, it is also imagined as one in which certain resources (disk space, computational power, memory) are valuable commodities that must be tightly regulated. Regulating space allows its productivity to be maximized.

What are bioinformaticians doing in virtual space? At the most literal level, they are usually interacting with screens of text – typing commands, writing programs, and moving or copying files. Almost always, they will be in more than one place at once – there is no limit to the number of computers to which a user can connect via *ssh,* and bioinformaticians are routinely logged into many at once, using a windows environment to flick back and forth between the various connections. For instance, a bioinformatician's task might involve simultaneously writing a program on his or her own machine, looking up a database on a public server, and copying data to or from disk space on a third machine. Like software engineers, a large part of the virtuosity of such work is in being able to move oneself and one's data rapidly between places. Indeed, bioinformaticians constantly seek to reduce the number of keystrokes necessary to move around. They can do this by setting up aliases, short commands that act as abbreviations of longer ones. Or, they can use their knowledge of programming languages such as Perl and regular expressions to short-cut themselves around all but the most intricate of maneuvers. In programs and on the command line it is common to see bioinformaticians using abstruse strings (for instance: “*{^(?:[^f]|f(?!oo))\*$}”*) in order to save themselves extra typing. Having a working grasp of such intricacies, combined with a knowledge of where important files and programs are located on the network, makes a skillful bioinformatician.

Much of the work of bioinformatics can be understood as the motion and transformation of data around virtual space. At EBI, I closely followed the progress of the “release cycle”, a process which occurs every couple of months through which the EBI’s main database (known as Ensembl) is revised and updated. A detailed description of the release cycle will illustrate the importance of space-management in bioinformatic work. Much of the work of the release coordinator is making sure that the right data ends up in the right place in the right form. Ensembl does not produce its own data – instead, its role it to collect data from a wide variety of sources and make it widely available in a common, coherent, and consistent format. Ensembl is also an automatic annotation system: it is software that takes raw genomic sequence and identifies the locations of particular structures (eg. genes) and their functions. Making a release involves collecting data from many individuals and places and running it through a software pipieline. For such a large set of databases it is not possible to simply update them one by one. Ensembl requires a sophisticated ‘staging’ system whereby the new release is prepared, processed, tested, and checked for consistency before it is released ‘live’ onto the World Wide Web. Thus the release cycle becomes a carefully choreographed set of motions through a virtual space in which data is migrated from the ‘live mirror’ to the ‘staging server’ and back again. I have excerpted here from the technical document used by the release coordinator:

**<space>**

**<alternate font for code>**

Firstly, check there is enough space for the new databases, e.g. in /mysql/data\_3306/databases/df -h /mysql

# Release 50: 1000G 342G 658G 35% /mysql

You should be able to see how much space is required here (may be out of date):

http://www.ensembl.org/info/webcode/requirements.html http://www.ensembl.org/info/webcode/installation/index.html

Release 50: Total required = 632.8 GB

Next you will need to modify the following script:

cd /ensembl-personal/release\_coordination/scripts

...

Now, run the script. For release 50:

cd /ensembl-personal/release\_coordination/scripts

perl get\_databases\_from\_ens-livemirror\_to\_ens\_staging.pl 50

The script generates an appropriate input file,

ens-livemirror\_to\_ens\_staging\_databases

to use with

ensembl/misc-scripts/CopyDBoverServer.pl

To actually copy the databases across from ens-livemirror to ens-staging you must be on ens-staging and logged in as mysqlens. Ask a previous release coordinator for the password if you don't already know it.

ssh ens-staging

su - mysqlens

Now run the copy script:

cd ensembl/misc-scripts/

perl CopyDBoverServer.pl -pass ensembl /path/to/file/ens livemirror\_to\_ens\_staging\_databases

Save and check the output from this script.[[36]](#footnote-36)

**<end alternate font>**

**<space>**

This describes in detail the procedures used to copy databases from the ‘live mirror’ to the ‘staging server.’ The crucial knowledge in this document is about *where* to find the various databases and the programs to move them. The ‘scripts’ are short Perl programs (for example, get\_databases\_from\_ens-livemirror\_to\_ens-staging.pl – the purpose of the program is made obvious by its name) that are used as short-cuts for copying large numbers of databases at once. In order to achieve the last step, the release coordinator must be “on” (that is, logged into) the staging server. The document, over forty pages long in all, provides step-by-step instructions on how to move oneself and the data around in virtual space order to perform the release cycle.

This thick description of the work at EBI shows the complexity of collecting, storing, and organizing biological data. Without such work, genomic biology would not be possible. Ensembl is an essential tool for managing big data, for making long strings of As, Gs, Ts, and Cs into “genes,” “regulatory elements,” and other biological objects. The Ensembl web-based interface attracts hundreds of thousands of users per month and it has been cited in hundreds of research articles.[[37]](#footnote-37) User-biologists can download short sequences or whole genomes, access specific data based on Ensembl’s annotations (for instance, just find particular genes), compare data between genomes, analyze intra-species variation, or examine regulatory sequences. What biologists can do with Ensembl is exactly the kind of thing biologists need to do all of the time: it is the very work of biology itself. For example, the following excerpt is from a website giving advice about how to use Ensembl:

Let’s say you have a set of genes in one species and you want to know the orthologs in another species and gene expression probes in that species you can use to assay those orthologs. For example, [given] 25 gene expression probes that are dysregulated in humans when exposed to benzene. What if you only had the U133A/B Affymetrix probe IDs and wanted to know the gene names? What if you also wanted all the Ensembl gene IDs, names, and descriptions of the mouse orthologs for these human genes? Further, what are the mouse Affymetrix 430Av2 probe IDs that you can use to assay these genes’ expression in mouse? All this can be accomplished for a list of genes in about 60 seconds using [Ensembl].[[38]](#footnote-38)

Ensembl’s tools allow biologists to rapidly deal with large amounts of data in different formats, comparing it across different organisms. It – and other tools like it – are the most valuable resources for studying genes and genomes, allowing biologists to manipulate and analyze the vast amount of available genomic data.

Bioinformatics is the task of ordering biological data in order to make it usable for biological research. Bioinformaticians must strive to maintain close control over their spaces, restricting access, protecting hierarchies and structures because it is the integrity of this space which determines the value of their work – in other words, it is through the organization of data in virtual space that bioinformation can become bioknowledge. Ensembl’s work should be understood as integral to the knowledge-making practice of biology. This data management or data curation consists of this work of organizing data into value. A disordered or haphazardly organized space would be of no value because it would contain no knowledge, reveal nothing about biology. Thus the way data is moved around in space and the way it is arrayed in space determines its epistemological status – data that is appropriately ordered in virtual space can attain the status of biological knowledge.[[39]](#footnote-39)

A great deal of effort and computational time is invested in keeping the data where it should be because the Ensembl brand depends on it: if information in the publicly accessible databases is to maintain a reputation for being reliable and usable knowledge it must remain highly ordered at all times. The Ensembl team itself conducts no experiments, produces no raw data directly from wet samples. Yet its work is highly valued by biologists because of its contribution to organizing data into knowledge. The particular ways in which Ensembl structures and organizes information are considered to be its trademark and its greatest asset. What I have shown here is how the careful management of and navigation through virtual space constitutes the work of bioinformatics knowledge-making. The flattening of both sequences and language described in sections 4.1 and 4.2 makes possible this informatic knowledge-making: it makes it possible to do biology in virtual space. This does not, however, make biology completely fluid. Rather, the constitution of biological objects becomes dependent on all sorts of structures build into hardware and software. By understanding networks and hard-drives as a space which can be traversed, surveyed, farmed, and mapped we can begin to understand how biology is constrained by the virtual spaces it inhabits.

*4.4 Conclusions*

This chapter has considered the virtual spaces in which biologists move around and through which they make knowledge. In contemporary biology, different kinds of spaces and different kinds of movements through them produce different kinds of knowledge. Information technologies require the flattening of biological objects and language into data. Pipelines, ontologies are ways of achieving this flattening. These structures open up new modes of knowledge creation through the motion of data in virtual space. But the kind of knowledge that emerges is dependent on and shaped by the hardware and software it subsists on: data structures, directed graphs, databases, servers, and compute farms.

These processes of flattening are also processes of making data fit for travel. The sequencing ‘pipeline,’ for instance, produces a particular kind of bioinformatic knowledge that is stripped of its messiness, contingency, and situatedness – this provides sequences with their peculiar ability to travel and recombine as fluid, universal objects. Much has been made of the fact that bioinformatic biology is globalizing. Many accounts suggest that biological objects are becoming place-free, homogenous, and fluid. But as far as biological work goes, this uniformity and liquidity is not automatic – this chapter has shown how this is an achievement, a consequence of biological work. The pipelines and ontologies and release cycles are ways of making heterogeneous and lumpy data into smooth, universal data that can travel with ease.

In other words, it is making the local appear universal. Understanding how biology has become and is becoming global or universal must begin with understanding how certified knowledge is produced in the labs and networks such that it can then travel through global spaces. Biotechnology, personalized genomics and so on seem to be playing out on a world-wide scale; however, my claim is that what is significant in these fields is how knowledge is made such that it *can* be globalized; how is it that a sequence produced at 320 Charles becomes *the* sequence of an elephant’s genome? In this chapter we have seen how it is movements through particular sorts of spaces, and in particular movement through the ‘sequencing pipeline,’ that constitute this sort of knowledge production: the transformation from a particular material sample to a universal (and universally-certified) digital representation takes places through a detailed and carefully policed series of steps acted out in space. The feature of globalization relevant to bioinformatics is not deterritorialization, but rather ‘interconnectedness’: the specificity of particular situations and conformations in particular laboratories and networks have a greater ability to make their influence felt everywhere.[[40]](#footnote-40) The Broad’s elephant becomes everyone’s elephant. In assessing the status and significance of ‘globalized’ biological knowledge, we need to continually recall its dependence on the situations from which it springs. Doing so will remind us that bioinformatic knowledge is never automatically fluid or universal, but always dependent on carefully constructed networks, structures, hierarchies, and sets of movements that keep it mobile.[[41]](#footnote-41)

The production of data entails a particular kind of working and a particular kind of knowing for biology. Central to this point is the idea that bioinformatics requires standardization – a kind of flattening of the biological object – in order to function. Moving objects around in virtual space means making them computable, networkable, and so on. Nucleotide sequences have become such standardizable, computable, networkable, objects. Making bioinformatics, then, has had much to do with constructing sequences in this way – as just such standard objects. The reducibility of sequence to data, to an object that can flow through the computer, has played a major role in establishing its importance and ubiquity in contemporary biological work. Sequence permits precisely the kind of abstraction or stripping down that is required for samples to transform into data. Bioinformatics has emerged ‘out of sequence’ because it is sequence that has made it possible to move biology around virtual space.

1. The science studies literature has emphasized the importance of physical and geographic spaces: see Collins, “TEA set,” Shapin, “Pump and circumstance.” Very little attention has been given to virtual spaces and proximities, however. Recently Collins has asked whether “electronic communication makes any difference to the nature of expertise?” (He concludes that it does not: see Collins, “Does electronic communication make any difference.”) [↑](#footnote-ref-1)
2. Vokoun, “Operations capability improvement,” 28-30. [↑](#footnote-ref-2)
3. Chang, “Control and optimization,” 12-14. [↑](#footnote-ref-3)
4. Person, “Operational streamlining,” 24. [↑](#footnote-ref-4)
5. Person, “Operational streamlining,” 28-29. [↑](#footnote-ref-5)
6. See [www.phrap.com](http://www.phrap.com/). Also see Ewing et al., “Base calling.” [↑](#footnote-ref-6)
7. In fact, any sequence which is worked on during a given day is submitted or re-submitted to GenBank at the end of that day; only finished sequences acquire ‘finished’ status in the database, however. [↑](#footnote-ref-7)
8. This point is made in detail in Barnes and Dupre, *Genomes*, 103-109. [↑](#footnote-ref-8)
9. For a more detailed account of this transition see Stevens, “Coding sequences.” [↑](#footnote-ref-9)
10. Dowell, “Distributed annotation system.” [↑](#footnote-ref-10)
11. Smith, “Ontology,” 155. [↑](#footnote-ref-11)
12. Smith, “Ontology,” 160. [↑](#footnote-ref-12)
13. Smith, “Ontology,” 162. [↑](#footnote-ref-13)
14. Lewis, “Gene ontology,” 104. [↑](#footnote-ref-14)
15. Lewis, “Gene ontology,” 104. [↑](#footnote-ref-15)
16. Lewis, “Gene ontology,” 104; interview with Michael Ashburner, 10 December 2008, Cambridge UK. [↑](#footnote-ref-16)
17. Ashburner, “On the representation of ‘gene function.’” Ashburner suggested linking GO to Stanford’s ‘Ontolingua’ project (<http://ksl-web.stanford.edu/knowledge-sharing/ontolingua/> ) and Schulze-Kremer’s ontology for molecular biology (Schulze-Kremer, “Ontologies for molecular biology”). [↑](#footnote-ref-17)
18. Interview with Michael Ashburner, 10December 2008, Cambridge UK. [↑](#footnote-ref-18)
19. Technically, GO is structured as a set of ‘directed acyclic graphs’ – like a tree hierarchy, but where each term can have multiple parents. For instance, if a particular gene was involved in making hexose, it’s GO terms would include ‘hexose biosynthetic process,’ ‘hexose metabolic process’ (since any biosynthetic process is (less specifically) a metabolic one) and ‘monosaccharide biosynthetic process’ (since hexose is (less specifically) a monosaccharide). Terms can be semantically linked to one another through relationships such as ‘is\_a’ or ‘regulates\_positively.’ GO conforms to the specifications of a Web Ontology Language (OWL) since its designers expect biological databases to be searched over the web (see the conclusion for more details on the relationship between bio-ontologies and web ontologies). [↑](#footnote-ref-19)
20. Gene Ontology Consortium, “Gene Ontology,” 25. [↑](#footnote-ref-20)
21. “The Consortium emphasized that GO was not a dictated standard. Rather, it was envisioned that groups would join the project because they understood its value and believed it would be in their interest to commit to the ontology. Participation through submission of terminology for new areas and challenging current representation of functionality was encouraged.” Bada, “Short study.” [↑](#footnote-ref-21)
22. Interview with Midori Harris, 26 November 2008, Hinxton UK. In fact there are four full time editors and a roughly further forty contributing researchers who have permission to write to (that is, edit) the GO Concurrent Versioning System. [↑](#footnote-ref-22)
23. OBO Foundry, “Open Biological and Biomedical Ontologies.” [↑](#footnote-ref-23)
24. One such ontology is called the Basic Formal Ontology. [↑](#footnote-ref-24)
25. Smith, “What is an ontology?” [↑](#footnote-ref-25)
26. See [http://www.flickr.com/groups/folksonomy/](http://www.flickr.com/groups/folksonomies/) Also see Ledford, “Molecular biology gets wikified.” [↑](#footnote-ref-26)
27. Smith, “What is an ontology?” [↑](#footnote-ref-27)
28. Smith, “What is an ontology?” [↑](#footnote-ref-28)
29. Smith, “Ontology as the core discipline.” [↑](#footnote-ref-29)
30. Leonelli, “Centralizing labels.” [↑](#footnote-ref-30)
31. For instance, the DAVID (Database for Annotation, Visualization and Integrated Discovery) web resource allows users to combine ontology terms and other identifiers from many different sources. See <http://david.abcc.ncifcrf.gov/> and Sherman, “DAVID knowledgebase.” The plausibility of GO becomes particularly important in high-throughput experiments in which tens or even hundreds of genes may be simultaneously up- or down-regulated by some exogenous conditioning (a drug, for example). The lab biologist then wishes to know whether these genes have anything in common. This can be done by testing for the ‘enrichment’ of annotation terms amongst the set of genes. For example, if 50 of the 60 up-regulated genes were annotated with the GO term ‘cell-cycle’ then the biologist may be able to conclude that the drug has an effect on the cell-cycle. [↑](#footnote-ref-31)
32. On the semantic web and data-sharing in biology see: Ure et al., “Aligning technical and social infrastructure.” [↑](#footnote-ref-32)
33. Field et al., “Minimum information.” [↑](#footnote-ref-33)
34. Again, how the machines are physically linked together in real space was of usually of little importance (unless something went wrong). Machines could be physically linked in complicated ways via routers, switches, virtual private networks, login hosts, load balancers, servers, and cache machines. [↑](#footnote-ref-34)
35. Sanger Center, “The Farm FAQ.” [↑](#footnote-ref-35)
36. European Bioinformatics Institute, “Ensembl release coordination.” [↑](#footnote-ref-36)
37. In 2011, Ensembl was accessed by about 3.4 million unique web (IP) addresses per year (<http://www.ebi.ac.uk/Information/News/press-releases/press-release-28112011-directors.html>). A quick search of PubMed yields about 450 articles that refer to Ensembl. [↑](#footnote-ref-37)
38. <http://gettinggeneticsdone.blogspot.com/2012/05/video-tip-use-ensembl-biomart-to.html> [↑](#footnote-ref-38)
39. In the discussion of databases in Chapter Five I examine how specific organizational structures affect the content of biological knowledge. Here, I am arguing that some sort of organization, some sense of spatial thinking, is necessary for granting ‘knowledge’ status to pieces of biological data in the first place. [↑](#footnote-ref-39)
40. The notion of interconnectedness comes from Tomlinson, *Globalization*. [↑](#footnote-ref-40)
41. For more on how knowledge travels see Morgan and Howlett, *How well do facts travel*. [↑](#footnote-ref-41)