We thank the reviewers for their time in providing comments and suggestions for our manuscript. We believe the revisions we have made in response have made the work stronger. The following is a point by-point summer of our responses (regular font) to each of the *reviewer comment (italics)*. The manuscript with tracked changes is attached to this document. Please note that some of the changes were made in response to editor comments.

***Reviewer #1:***

*Accept*

We thank the reviewer for his/her evaluation of our manuscript.

***Reviewer #2:***

*Although the paper is clear, well written and I completely agree with the basic idea of the importance in exploiting ICT resources for improving tissue engineering practices, my opinion is that the described protocol is not completely effective for the purpose. The protocol is too simplistic for being useful and exploitable to solve real issues of the field, while the description is not deep enough for enabling non-expert users to take advantage from it. In summary, the idea is entirely acceptable and well explained but the way the protocol is presented seems not suitable to help tissue engineering data storage and management issues.*

We thank the reviewer for this evaluation. We have adjusted the protocol to provide some simpler instructions. We also provide the software that is discussed in the manuscript, which will allow the users to run it on their own with simply substituting their data for what was used here. We believe that in this way, there is a balance between presenting something that is comprehensive enough to be useful without overwhelming the novice user with specific terminology.

***Reviewer #3:***

*Comment:*

*Given that this is an invited topic for a special issue, it is apparent that the JoVE editors are interested in such an article. Despite years of preaching the relatively simple idea that this article is espousing, it is a little depressing that such an article is even necessary.*

We thank the reviewer for his/her comment, and we believe that this will be indeed useful to many researchers who have not heard of this method of handling data or do not know how to get started.

*Minor Concerns:*

*There may be a mismatch between the "high" level of most of the publication vs the much more detailed level of specifying SQL statements. Are there other ways to get reports out of the database other than writing SQL statements such as these? Such a technical portion of the paper may scare off the intended audience.*

We have added an alternative instruction on how to create simple queries through the graphical interface, but we found that for many users who are familiar with matlab programming, the SQL instructions were simpler to grasp than the graphical interface. Thus we will leave both methods in the protocol.

***Reviewer #4:***

*The paper will benefit from addressing the following points.*

*Major Concerns:*

*-Explain the term low velocity; this can be not familiar to the broad scientific audience.*

An explanation friendly to the broader scientific audience has been provided (2nd paragraph of Introduction).

*-Provide more examples of the typical medium sized, low velocity, multidimensional data in the research.*

More examples have been provided in the text.

*-The paper focused on relational databases; however, it will be beneficial to include a discussion about the non-relational database such as NoSQL and describe the pros and cons.*

Thank you for this good suggestion. A discussion about non-relational databases and its comparison to relational databases has been added.

*-Suggest strategies to make sure that path to the file, included in the database, remains unchanged.*

Several strategies are discussed in Protocol Section 4.2.

*-It is essential to make emphasis that raw data should be preserved.*

The statement emphasizing that raw data should be left untouched has been added to Protocol section 3.

*-To introduce to best practices in the writing analysis code, it is worth mentioning version control approaches, for example, via the git.*

This has been added to Protocol section 4.1.

*-The community will benefit from a discussion of the best practices for folder/subfolder structure and file names, at least refer to the corresponding resources.*

Commentary on good naming convention has been added to section 2.1.

*-Emphasize the importance of the readme document that describes the structure of the database.*

This has been added to the other good practice discussions in section 2.3.

*Minor Concerns:*

*-Mention the version of the Matlab that was used to write the code.*

The version number has been added to the Materials table.

*-Include a mathematical definition of the orientational order parameter (OOP) in the Methods section.*

The definition has been added along with the references that go into more of a description.

***Reviewer #5:***

*Major Concerns:*

*I dont have any major concerns*

*Minor Concerns:*

**Why author used microfluidic system to grow the fibroblast cells, it could be easily done in 2D and 3D system? In 2D and 3D system they can collect huge amount of data and by implement RDBMS they can organize the data in more specific way.*

We agree with the reviewer. Indeed, this project does not use microfluidics. The fibroblasts were cultured on 2D cover-slips that were either isotropic or patterned with the use of micro-contact printing of extracellular matrix (as described in the papers referenced in the methods section). The data organization is flexible in that it can be used with any method that collects the multi-dimensional data described in this work as simply an example.

**This manuscript is based on database management system and author took Progeria as an example, it should be good to include some more information related to Progeria such as free radicals and reactive oxygen species production which play a major role in etiology and progression of neurodegenerative disease like progeria.*

We agree that in studying Progeria, there are many other interesting assays that can be run. However, in this example Progeria was used solely as a positive control to the family cell lines that have a mutation to the same gene. Because of the large number of cell-lines, the experiments described are optimally suited as an example for this data-base introduction JoVe video.

**A comparative data related to oxidative stress between the progeric and normal fibroblast should be better to included.*

We agree that the oxidative stress comparison between mutated and non-muted lines could be an interesting data-set, but it is far outside the scope of the experimental work where Progeria is a positive control. It might be interesting to add to another work where the functional differences are the main target of the paper rather than handling of multi-dimensional data.

**TITLE:** Databases to Efficiently Manage Medium Sized, Low Velocity, Multidimensional Data in Tissue Engineering

**AUTHORS AND AFFILIATIONS:** Alexander R. Ochs[1,2], Mehrsa Mehrabi[1,2], Danielle Becker[1,2], Mira N. Asad[1,2], Jing Zhao[1,2], Michael V. Zaragoza[3,4], Anna Grosberg[1,2,5,6,7*]

1 Department of Biomedical Engineering, University of California, Irvine, CA, United States of America

2 The Edwards Lifesciences Center for Advanced Cardiovascular Technology, University of California, Irvine, CA, United States of America

3 Pediatrics-Genetics & Genomics Division-School of Medicine, University of California, Irvine, CA, United States of America

4 Biological Chemistry-School of Medicine, University of California, Irvine, CA, United States of America

5 Department of Chemical and Biomolecular Engineering, University of California, Irvine, CA, United States of America

6 Center for Complex Biological Systems, University of California, Irvine, CA, United States of America

7 The NSF-Simons Center for Multiscale Cell Fate Research (CMCF), University of California, Irvine, CA, United States of America

* Corresponding author
Contact Information:
Alexander R. Ochs < ochsa@uci.edu>
Mehrsa Mehrabi < mehrabim@uci.edu>
Danielle Becker < mbecker@uci.edu>
Mira N. Asad < mnasad@uci.edu>
Jing Zhao < jingzhao022@gmail.com>
Michael V. Zaragoza < mzaragoz@uci.edu>
Anna Grosberg <
~~Email:~~ grosberg@uci.edu>

**KEYWORDS:** Medium sized data; databases; LMNA; data organization, multidimensional data, tissue engineering

**SUMMARY:** Many researchers generate “medium-sized,” low-velocity, and multi-dimensional

data, which can be managed more efficiently with databases rather than spread-sheets. Here we provide a conceptual overview of databases including visualizing multi-dimensional data, linking tables in relational database structures, mapping semi-automated data pipelines, and using the database to elucidate data meaning.

**ABSTRACT:**
Science relies on increasingly complex data sets for progress, but common data management methods such as spreadsheet programs are inadequate for the growing scale and complexity of this information. While database management systems have the potential to rectify these issues, they are not commonly utilized outside of business and informatics fields. Yet, many research labs already generate "medium sized," low velocity, multi-dimensional data that could greatly benefit from implementing similar systems. In this article, we provide a conceptual overview explaining how databases function and the advantages they provide in tissue engineering applications. Structural fibroblast data from individuals with a Lamin A/C mutation was used to illustrate examples within a specific experimental context. Examples include visualizing multidimensional data, linking tables in a relational database structure, mapping a semi-automated data pipeline to convert raw data into structured formats, and explaining the underlying syntax of a query. Outcomes from analyzing the data were used to create plots of various arrangements and significance was demonstrated in cell organization in aligned environments between the positive control of Hutchinson-Gilford Progeria, a well-known laminopathy, and all other experimental groups. In comparison to spreadsheets, database methods were enormously time efficient, simple to use once set up, allowed for immediate access of original file locations, and increased data rigor. In response to the NIH emphasis on experimental rigor, it is likely that many scientific fields will eventually adopt databases as common practice due to their strong capability to effectively organize complex data.

**INTRODUCTION:**
In an era where scientific progress is heavily driven by technology, handling large amounts of data has become an integral facet of research across all disciplines. The emergence of new fields such as computational biology and genomics underscores how critical the proactive utilization of technology has become. These trends are certain to continue due to Moore's law and steady progress gained from technological advances [1,2]. One consequence, however, is the rising quantities of generated data that exceed the capabilities of previously viable organization methods. Although most academic laboratories have sufficient computational resources for handling complex data sets, many groups lack the technical expertise necessary to construct custom systems suited for developing needs [3]. Having the skills to manage and update such data sets remains critical for efficient workflow and output. Bridging the gap between data and expertise is important for efficiently handling, re-updating, and analyzing a broad spectrum of multifaceted data.

Scalability is an essential consideration when handling large data sets. "Big data," for instance, is a flourishing area of research that involves revealing new insights from processing data characterized by huge volumes, large heterogeneity, and high rates of generation, such as audio and video [4,5]. Using automated methods of organization and analysis is mandatory for this field to appropriately handle torrents of data. Many technical terms used in big data are not clearly defined, however, and can be confusing; for instance, "high velocity" data is often associated with millions of new entries per day whereas "low velocity data" might only be hundreds of entries per day, such as in an academic lab setting. Although there are many exciting findings yet

to be discovered using big data, most academic labs do not require the scope, power, and complexity of such methods for addressing their own scientific questions [5]. While it is undoubtable that scientific data grows increasingly complex with time [6], many scientists continue to use methods of organization that no longer meet their expanding data needs. For example, convenient spreadsheet programs such as Microsoft (MS) Excel are frequently used to organize scientific data, but at the cost of being unscalable, error prone, and time inefficient in the long run [7,8]. Conversely, databases are an effective solution to the problem as they are scalable, relatively cheap, and easy to use in handling varied data sets of ongoing projects.

Immediate concerns that arise when considering schemas of data organization are cost, accessibility, and time investment for training and usage. Frequently used in business settings, database programs are more economical, being either relatively inexpensive or free, than the funding required to support use of big data systems. In fact, a variety of both commercially available and open source software exists for creating and maintaining databases, such as Oracle Database, MySQL, and MS Access [9]. Many researchers would also be encouraged to learn that several MS Office academic packages come with MS Access included, further minimizing cost considerations. Furthermore, nearly all developers provide extensive documentation online and there is a plethora of free online resources such as Codecademy, W3Schools, and SQLBolt to help researchers understand and utilize structured query language (SQL) [10-12]. Like any programming language, learning how to use databases and code using SQL takes time to master, but with the ample resources available the process is straightforward and well worth the effort invested.

Databases can be powerful tools for increasing data accessibility and ease of aggregation, but it is important to discern which data would most benefit from a greater control of organization. Multi-dimensionality refers to the number of conditions that a measurement can be grouped against, and databases are most powerful when managing many different conditions [13]. Conversely, information with low dimensionality is simplest to handle using a spreadsheet program such as MS Excel; for example, a data set containing years and a value for each year has only one possible grouping (measurements against years). High dimensional data such as from clinical settings would require a large degree of manual organization in order to effectively maintain, a tedious and error-prone process beyond the scope of spreadsheet programs [13].-Non-relational (NoSQL) databases also fulfill a variety of roles, primarily in applications where data does not organize well into rows and columns [14]. In addition to being frequently open source, these organizational schema include graphical associations, time series data, or document-based data. NoSQL excels at scalability better than SQL, but cannot create complex queries, so relational databases are better in situations that require consistency, standardization, and infrequent large-scale data changes [15]. Databases are best at effectively grouping and re-updating data into the large array of conformations often needed in scientific settings [13,16].

The main intent of this work, therefore, is to inform the scientific community about the potential of databases as scalable data management systems for “medium sized,” low velocity data as well as to provide a general template using specific examples of patient sourced cell-line experiments. Other similar applications include geospatial data of river beds, questionnaires from longitudinal clinical studys, and microbial growth conditions in growth media [17-19]. This work highlights

common considerations for and utility of constructing a database coupled with a data-pipeline necessary to convert raw data into structured formats. The basics of database interfaces and coding for databases in structured query language (SQL) are provided and illustrated with examples to allow others to gain the knowledge applicable to building basic frameworks. Finally, a sample experimental data set demonstrates how easily and effectively databases can be designed to aggregate multifaceted data in a variety of ways. This information provides context, commentary, and templates for assisting fellow scientists on the path towards implementing databases for their own experimental needs.

## PROTOCOL

**METHODS ~~NOT PART OF PROTOCOL~~ – NEEDED FOR EXAMPLE DATA-SET USED FOR RESULTS:**

For the purposes of creating a scalable database in a research laboratory setting, data from experiments using human fibroblast cells was collected over the past three years. The primary focus of this protocol is to report on the organization of computer software to enable the user to aggregate, update, and manage data in the most cost- and time-efficient manner possible, but the relevant experimental methods are provided as well for context.

### Experimental Setup

The experimental protocol for preparing samples has been described previously [20,21], and is presented briefly here. Constructs were prepared by spin-coating rectangular glass coverslips ~~(Fisher Scientific Company, Hanover Park, IL)~~ with a 10:1 mixture of polydimethylsiloxane (PDMS~~, Ellsworth Adhesives, Germantown, WI~~) and curing agent, then applying 0.05 mg/mL fibronectin ~~(Corning, Corning, NY)~~, in either unorganized (Isotropic) or 20 µm lines with 5 µm gap micropatterned arrangements (Lines). Fibroblast cells were seeded at passage 7 (or passage 16 for positive controls) onto the coverslips at optimal densities and left to grow for 48 hours with media being changed after 24 hours. The cells were then fixed using 4% paraformaldehyde (PFA) solution ~~(Fisher Scientific Company, Hanover Park, IL)~~ and 0.0005% Triton-X ~~(Sigma-Aldrich Inc., Saint Louis, MO)~~, followed by the coverslips being immunostained for cell nuclei (4',6'-diaminodino-2-phenylinodole (DAPI)~~, Life Technologies, Carlsbad, CA~~), actin (Alexa Fluor 488 Phalloidin~~, Life Technologies, Carlsbad, CA~~), and fibronectin (polycloncal rabbit anti-human fibronectin~~, Sigma Aldrich Inc., Saint Louis, MO~~). A secondary stain for fibronectin using goat anti-rabbit IgG antibodies (Alexa Fluor 750 goat anti-rabbit~~, Life Technologies, Carlsbad, CA~~) was applied and Prolong Gold Antifade ~~(Life Technologies, Carlsbad, CA)~~ was mounted onto all coverslips to prevent fluorescent fading. Nail polish was used to seal coverslips onto microscope slides then left to dry for 24 hours.

Fluorescence images were obtained as described previously [20] using an ~~UPLFLN~~ 40x oil immersion objective ~~(Olympus America, Center Valley, PA)~~ coupled with a digital CCD camera ~~ORCAR2 C10600-10B (Hamamatsu Photonics, Shizuoka Prefecture, Japan)~~ mounted on an ~~IX-83~~ inverted motorized microscope ~~(Olympus America, Center Valley, PA)~~. Ten randomly selected fields of view were imaged for each coverslip at 40x magnification, corresponding to a 6.22 pixels/µm resolution. Custom-written ~~Matlab~~ codes were used to quantify different variables from the images describing the nuclei, actin filaments, and fibronectin; corresponding values, as well as organization and geometry parameters, were automatically saved in ~~Matlab~~ data files.

**Cell Lines**

More extensive documentation on all sample data cell lines can be found in prior publications [20]. To describe briefly, the data collection was approved and informed consent was performed in accordance with UC Irvine Institutional Review Board (IRB # 2014-1253). Human fibroblast cells were collected from three families of different variations of the Lamin A/C (*LMNA*) gene mutation: heterozygous *LMNA* splice-site mutation (c.357-2A>G) [22] (Family A); *LMNA* nonsense mutation (c.736 C>T, pQ246X) in exon 4 [23] (Family B); and *LMNA* missense mutation (c.1003C>T, pR335W) in exon 6 [24] (Family C). Fibroblast cells were also collected from other individuals in each family as related mutation-negative controls, referred to as "Controls," and others were purchased as unrelated mutation-negative controls, referred to as "Donors." As a positive control, fibroblast cells from an individual with Hutchinson-Gliford Progeria (HGPS) were purchased and grown from a skin biopsy taken from an 8-year-old female patient with HGPS possessing a *LMNA* G608G point mutation [25]. In total, fibroblasts from 22 individuals were tested and used as data in this work.

**Data Types**

Fibroblast data fell into one of two categories: cellular nuclei variables (i.e., percentage of dysmorphic nuclei, area of nuclei, nuclei eccentricity) [20] or structural variables stemming from the orientational order parameter (OOP) [21,26,27] (i.e., actin OOP, fibronectin OOP, nuclei OOP). This parameter is equal to the maximum eigenvalue of the mean order tensor of all the orientation vectors, and it is defined in detail in previous publications [26,28]. These values are aggregated into a variety of possible conformations, such as values against age, gender, disease status, presence of certain symptoms, etc. Examples of how these variables are used can be found in the Results section.

**Software**

~~Matlab R2018b (Mathworks, Natick, MA) was used as a coding software to create a data-pipeline and MS Access (Microsoft, Redmond, WA) was used as a database software. Queries were created in MS Access using structured query language (SQL) to specify requests for information via aggregation.~~Please see the Materials Table for the software versions used in this protocol.

**PROTOCOL:**

1. Evaluate if your data would benefit from a database organization scheme. ~~The first step when considering the use of databases is to evaluate if the data would benefit from such an organization.~~

1.1 Download the example codes and databases.

~~1.1~~ Use Figure 1 to ~~E~~evaluate if ~~the~~your data-set is "multi-dimensional." Figure 1 is a graphical representation of a multi-dimensional database provided for the example data-set.

1.2 ~~In the context of this example the subjects, described in the Methods section, were divided into groups of individuals from the three families with the heart disease-causing *LMNA* mutation ("Patients"), related non-mutation negative controls ("Controls"), unrelated non-mutation negative controls ("Donors"), and an individual with Hutchinson-Gilford progeria syndrome (HGPS) as a positive control [20]. Results from Controls and Donors could be further grouped together as an overall Negative Control (N.C.) group, given their collective lack of *LMNA* mutations. Every subject's cell line had a "Mutation Status" associated with it, based on their condition group (Fig 1 – dark blue axis). For each experiment, fibroblast cells from the subjects were cultured on arrangements of either unorganized (Isotropic) or micropatterned (Lines) fibronectin, creating the condition of "Pattern type" (Fig 1 – orange axis). After the cells were fixed, immunostained, and imaged, the "Coverslip #" was transcribed, since multiple experiments (i.e., technical replicates) would occur using the same individual's cells (Fig 1 – light green axis). Custom Matlab codes [20,21] were then used to quantify different aspects of cell nuclei or tissue organization variables as "Variable type" (Fig 1 – teal green axis). The three factors were associated with the cells' human source and consequently linked to the "Family" (Fig 1 – dark pink axis) and "Age at time of biopsy" (Fig 1 – dark green axis) in addition to "Mutation Status." Other dimensions not included in Fig 1 were the "Age of presentation," "Symptoms," "Designator," and "Gender" of the individual in question.~~

~~1.3~~ If the data can be visualized in a "multi-dimensional" form like the example and if your ~~being~~ ability~~le~~ to relate a specific experimental outcome to any of the dimensions (i.e. conditions) would allow for greater scientific insight into the available data, ~~there is a benefit to~~proceed to construct~~ing~~ a relational database.

~~1.4~~1.3 ~~The example provided here results in at least ten possible dimensions for data aggregation. Thus this example data is a prime candidate for organization by relational databases.~~

**2. Organize your ~~the~~ database structure**

2.1 Relational databases store information in the form of tables. Tables are organized in schema of rows and columns, similar to spreadsheets, and can be used to link identifying information within the database. Thus, it is imperative for you to organize the data into a well thought out set of connected tables. Good practice with file naming conventions and folder-subfolder structures, when done well, allow for broad database scalability without compromising the readability of accessing files manually. Adding date files in a consistent format, such as "20XX-YY-ZZ", and naming subfolders according to metadata is one such example.

2.2 As you design the data-base structure, draw relationships between the fields in different tables. Thus you ~~H~~handle multi-dimensionality by relating different fields (i.e. columns in the tables) in individual tables to each other.

For this ~~example~~example, ~~by, for instance,~~ the differing characteristics of individuals (Fig 2A) are related to associated experimental data of those individuals (Fig 2B). The same was done through relating columns of Pattern Types (Fig 2C) and Data Types (Fig 2D) to matching entries in the main data values table to explain various shorthand notations (Fig 2B).

2.3 Create Readme documentation that describes the database and relationships you have created. Once an entry between different tables is linked, all associated information is related to that entry and can be used to call complex queries to filter down to the desired information. Readme documents are a common solution for providing supplemental information and database structural information about a project without adding non-uniform data to the structure.

3. Determine all the essential and merely helpful data points that need to be recorded for long range data collection

NOTE: A key advantage of using databases over spreadsheet programs, as mentioned earlier, is scalability: additional data points can be trivially added at any point and calculations, such as averages, are instantly updated to reflect newly added data points. However, it is important to identify the necessary information for creating distinct data points prior to beginning. Good practice critically requires that raw data is left untouched, instead of modifying or saving over it, so that reanalysis is possible and accessible.

For the given example, the Designator corresponding to an individual, Pattern type, Coverslip #, and Variable type were all vital fields for distinctness of the associated value. Other helpful, non-vital information can also be added such as the Total # of Coverslips to indicate the number of repetitions conducted and help determine if data points are missing.

**4. Setup and organize the pipeline**

Up to an estimated 95% of all digital data is unstructured, but structured formats are required for databases. Still, creating a good automated method for the data-pipeline is highly context dependent.

~~4.1~~ Identify all the various experiments and data analysis methods that might lead to data collection along with the normal data storage practices for each data type. Working with open source version control software such as GitHub also ensures necessary consistency and version control while minimizing user burden.

~~4.2~~4.1 ~~For this example, the images collected from each experiment were stored in folders named by date and initial of the lab member responsible, with sub folders listing the subject and coverslip number. Pipeline files are provided in the Supplemental Materials section, as well as summarized in a flow chart illustration (Fig 3). Different metrics from various experimental conditions across a variety of subjects were quantified from these fluorescent images (Fig 3A) using custom Matlab codes (Fig 3B)~~ [20,21]~~. For example, actin orientational order parameter~~ [21]~~ was extracted from tissues stained with phalloidin (Fig 3A) and used to compare the organization of fibroblasts from different individuals. The code outputs were saved in the same folder as the source images (Fig 3C).~~

~~4.3~~4.2 If possible, create procedure for consistent naming and storing of data to allow for an automated pipeline.

In the example, outputs were all consistently named, thus creating a data-pipeline that looked for specific attributes was straightforward once the files were selected.

*N~~ote~~OTE:* If consistent naming is not possible, the tables in the database will need to be

populated manually (NOT RECOMMENDED).

~~4.4~~4.3 Use any convenient programming language to generate new data entries for the database. Examples of programming languages can include MatLab, visual basic, or excel macros.

~~4.4.1~~4.3.1 One option is to create small "helper" tables in separate files that can guide automated selection of data. These files serve as a template of possibilities for the pipeline to operate under and are easy to edit.

In this example, to generate new data entries for the data-pipeline (Fig 3D), a Matlab code was run and three MS Excel files were selected by the user as inputs containing information for the different cell lines, variable types, and pattern types of the data, which is the information also located in the supporting non-value tables (Figs 2A, 2C-2D).

~~4.4.2~~4.3.2 Create an automated code that will ask the minimum input from the user and generate the table data structure.

In the example: The user entered the category of data type (cell nuclei or structural measurements), cell lines' subject designator, and number of files being selected. The relevant files were then selected by the user (Table 1, column 1), with the row entries being automatically created and populated with all variables contained within the file (Table 1, column 2).

~~4.4.3~~4.3.3 Make sure the code is flexible so that if another experimental entry needs to be added, the user can select to continue the loop; if not, the files are saved and the loop ends.

~~4.4.4~~4.3.4 From here a new spreadsheet of file locations should be assembled by combining the new entries with the previous entries (Fig 3E).

~~4.4.5~~4.3.5 Afterwards, this merged spreadsheet needs to be checked for duplicates, which should be automatically removed

~~4.4.6~~4.3.6 Additionally check the spreadsheet for errors, and ~~notifie~~notify the user of their reason and location (Fig 3F).

~~4.4.7~~4.3.7 Then use the file locations ~~should then be used~~ to generate a data values spreadsheet (Fig 3G) as well as to create a most updated list of entries that can be accessed to identify file locations or merged with future entries (Fig 3H).

NOTE~~ote~~: The basic functions of adding new entries, checking for errors, and assembling the spreadsheet from file locations are all critical for an efficient data-pipeline setup.

5. Double check that your pipe-line adds to the experimental rigor.

NOTE: It is imperative to note that using file locations when creating the data-pipeline increases experimental rigor. Specifically, having a corresponding spreadsheet listing all file locations for the data values allows a user to backtrack any data point back to the lab notebook of the researcher who collected the raw data. When dealing with hundreds to tens of thousands of data points, greater transparency and accessibility is invaluable over the lifetime of a project. We highly recommend users consider saving file locations first and later compiling values for data instead of only storing the data values.

6. **Create MS Access SQL Queries**

If tables store information in databases, then queries are requests to the database for information given specific criteria.

6.1 Open the database file that was downloaded earlier. The simplest way to get started is by

programming the queries through the design view ~~of MS Access~~. The user will find it useful to download the provided MS Access template as a starting point.

6.1.1 Select Create → 'Query Design'

6.1.2 Drag all relevant tables into the top window. In this example 'Cell Lines', 'Data Values', 'Data Types', and 'Pattern Type'

6.1.3 The relationships should automatically set-up based on the previous Relationship design.

~~6.1.0~~6.1.4 Fill out the query fields following the example provided in any of the queries (for example "Actin_OOP_Averages").

6.2 Alternatively, ~~queries are~~ code~~d~~ a query using SQL: Fig 4 shows a sample query using SQL syntax that is designed to run using the database relationships shown in Fig 2.

6.2.1 SQL usually requires SELECT (Fig 4A) and FROM (Fig 4B) statements to denote which tables and fields to use in a query.

6.2.2 Calculations such as averages, standard deviations, and counts can also be performed on the data within the SELECT statement (Fig 4A).

6.2.3 Additional criteria can be added in HAVING, and GROUP BY (Fig 4C and 4D) statements to allow for selection from the data.

6.2.4 ORDER BY in comparison simply lists the query outputs in a fashion better organized to the user's needs (Fig 4E).

7. Move ~~T~~the output tables ~~should be moved~~ to a statistical software for significance analysis

For this sample experimental data, the one-way Analysis of Variance (ANOVA) using Tukey's test was utilized for mean comparisons between various conditions. Values of $p < 0.05$ were considered statistically significant.

## REPRESENTATIVE RESULTS:

### Multi-dimensionality of the Data:

In the context of the~~is~~ example data-set presented here, the subjects, described in the Methods section, were divided into groups of individuals from the three families with the heart disease-causing *LMNA* mutation ("Patients"), related non-mutation negative controls ("Controls"), unrelated non-mutation negative controls ("Donors"), and an individual with Hutchinson-Gilford progeria syndrome (HGPS) as a positive control [20]. Results from Controls and Donors could be further grouped together as an overall Negative Control (N.C.) group, given their collective lack of *LMNA* mutations. Every subject's cell line had a "Mutation Status" associated with it, based on their condition group (Fig 1 – dark blue axis). For each experiment, fibroblast cells from the subjects were cultured on arrangements of either unorganized (Isotropic) or micropatterned (Lines) fibronectin, creating the condition of "Pattern type" (Fig 1 – orange axis). After the cells were fixed, immunostained, and imaged, the "Coverslip #" was transcribed, since multiple experiments (i.e., technical replicates) would occur using the same individual's cells (Fig 1 – light green axis). Custom Matlab codes [20,21] were then used to quantify different aspects of cell nuclei or tissue organization variables as "Variable type" (Fig 1 – teal green axis). The three factors were associated with the cells' human source and consequently linked to the "Family" (Fig 1 – dark pink axis) and "Age at time of biopsy" (Fig 1 – dark green axis) in addition to "Mutation Status." Other dimensions not included in Fig 1 were the "Age of presentation," "Symptoms," "Designator," and "Gender" of the individual in question. The example provided here results in at least ten possible dimensions for data aggregation. Thus this example data is a prime candidate for organization by relational databases.

### Organizing the Pipeline

For this example, the images collected from each experiment were stored in folders named by date and initial of the lab member responsible, with sub-folders listing the subject and coverslip number. Pipeline files are provided in the Supplemental Materials section, as well as summarized in a flow chart illustration (Fig 3). Different metrics from various experimental conditions across a variety of subjects were quantified from these fluorescent images (Fig 3A) using custom Matlab codes (Fig 3B) [20,21]. For example, actin orientational order parameter [21] was extracted from tissues stained with phalloidin (Fig 3A) and used to compare the organization of fibroblasts from different individuals. The code outputs were saved in the same folder as the source images (Fig 3C).

### Identifying A Novel Relationship in *LMNA* Mutation Data Set

When given multitude of possible conformations, it can be difficult to identify where novel relationships exist using manual data aggregation methods. In this specific context, we were interested in comparing the organization of subcellular actin filaments across multiple conditions, measured using the Orientational Order Parameter (OOP) [27]. OOP is a mathematical construct quantifying the degree of order in anisotropic environments, normalized to zero corresponding to completely isotropic tissue and one corresponding to completely aligned tissue. The data set was first split up by Pattern type as Lines (Fig 5A) and Isotropic (Fig 5B) conditions, which were expected to have vastly different OOPs since fibronectin micropatterning heavily influences tissue organization. There were no significant differences between conditions when comparing isotropic tissues (Fig 5B). Conversely, the patterned tissues were statistically less organized in the positive control cell line (HGPS) (Fig 5A), and this relationship held even when the data was

aggregated into different groups (Fig 5C). Actin OOP was additionally plotted against individuals' age at time of biopsy (Fig 5D), separated by mutation status and family, to illustrate aggregation against a clinical variable. Unlike with nuclear defects [20], there is no correlation between actin organization and an individual's age (Fig 5D). Ultimately, the plots shown in Fig 5 illustrate how the same data can be analyzed in different combinations and how easily the normally difficult task of aggregating data that falls under multiple classes can be accomplished using databases.

For this manuscript, data from patient sourced fibroblasts were compared between conditions to determine mutation consequences. Although both HGPS and the three families in this study have *LMNA*-linked diseases that potentially disrupt the nuclear envelope, the patients exhibit symptoms primarily associated with heart dysfunction whereas HGPS individuals have multiple organ systems affected [22-24]. Indeed, despite the micropatterned environment cells originating from an HGPS patient had a statistically lower actin OOP value than any of the other cell lines considered (Figs 5A and Fig 5C). This dovetails with HGPS patients being the only ones in the study with any skin abnormalities caused by the mutation. Viewing the same data in different conformations is also helpful for providing additional insight and avenues into scientific inquiry in a varied data set (Fig 5).

**FIGURE AND TABLE LEGENDS:**

**Fig 1**. **A visualization of multi-dimensional data from the *LMNA* Mutation data set**.
A single cube is defined by the three dimensions of "Variable type," "Pattern type," and "Coverslip #." Further dimensions are shown as the axes of "Mutation Status," "Age of biopsy" (yrs), and "Family." Colored labels correspond to the different axes shown, such as the Age of biopsy (green numbers) for each individual's cube. Here, six of the ten possible dimensions are used to illustrate the multi-dimensionality of experimental data points.

**Fig 2**. **Table and Design View relationships within the *LMNA* Mutation data set**.
Relational databases have the advantage of linking fields in one table with information in another table, which allows for immediate interchangeability of aggregation. The example here visually demonstrates how differing information can be linked.

**Fig 3**. **An example of common data-pipeline needs in a generalized context**.
New entries were created using user inputs and automated codes, formatting important information into a spreadsheet format. These entries were combined with the most recent set of file location entries, checked for errors, then stored as both a spreadsheet of file locations and a spreadsheet of data values. Scale bar = 20 μm

**Fig 4**. **An example query using SQL syntax**.
SELECT and FROM statements are requirements to generate a query, but additional commands and criteria are often included. GROUP BY provides clarification on how aggregate the data, HAVING or WHERE statements limit the output to data that meets specific criteria, and ORDER BY indicates the order by which the outputs should be arranged by.

**Fig 5. Comparisons between conditions for the actin OOP variable.**
(A)-(B): groupings correspond to the four primary conditions: non-related negative control Donors, related negative control Controls, *LMNA* mutation Patients from three families, and positive control HGPS. (C): all Negative Controls (N.C.) were combined and Patients were separated by family (PA, PB, PC) instead. D) demonstrates a potential graph of isotropic actin OOP against age at time of biopsy collected for this study, separated by condition and family. (A), (C), and (D) are plotted for the tissues micropatterned with a Lines pattern, while (B) is plotted for isotropic tissues. Statistical significance of $p<0.05$ (*) was found in (A), (C), and (D). No significance between any pairs was found in (B). All error bars represent standard deviations calculated within the Access database.

**Table 1: Listed select files that correspond to different variables of either cell nuclei measurements or fibroblast structural (OOP) data.**

**DISCUSSION:**

**Technical Discussion of the Protocol, Step-by-Step**

1. The first step when considering the use of databases is to evaluate if the data would benefit from such an organization.

**Scientific Discussion**

The purpose of this manuscript was to disseminate methods involving a data-pipeline and database that elucidated data set scalability and transparency. These methods are not widely used outside of informatics and business, but have enormous potential for those working in biological contexts. As science continues to rely on computers more heavily, the importance of effective management systems also rises [6,29]. Databases are frequently used for high volume and/or high velocity applications and are well cited in the literature, especially regarding their usage for clinical patient populations [8,30,31]. Several have already been constructed for specific fields such as the Rat Genome Database curation tools or REDCap for clinical and translational research [32,33]. Thus, the use of databases has been adopted in the clinical domain [8] or large genomic databases [32], but has not become common in other scientific disciplines such as tissue engineering.

The issues of handling increasingly complex data using spreadsheet programs have long been acknowledged within the scientific community [34]. One study reported that around 20% of genomic journal papers with supplemental files had gene names that were erroneously

converted to dates [35]. These mistakes increased at an average of 15% per year from 2010 to 2015, far outpacing the annual increase of genomics papers at 4% per year. It is often nearly impossible to identify individual errors within a large volume of data, as by nature spreadsheet programs are unsuited for easy validation of results or formula calculations. Published articles even exist for educating scientists on better spreadsheet practices in an attempt to reduce the frequency of errors [7]. One of the strongest benefits of databases is the reduction of error through automated methods and ability to validate potentially questionable data (Fig 3).

A significant outcome of this methodology is the increased rigor of data analysis. The importance of increasing the reproducibility of data has been highlighted by the NIH as well as by other scientists and institutions [36,37]. By having a spreadsheet of file locations corresponding to every database, it is easy to trace a data point back to the lab notebook of the experiment in question (Fig 3). Individual data points can also be quickly identified and found electronically using the corresponding file locations, which is invaluable at times, even when coupled with automatic error screening during the data-pipeline process. Even as the data set is amended over time, best practice involves keeping all past files in case issues occur or older versions need to be checked. Working non-destructively and keeping old versions within the data-pipeline creates security through redundancy and allows for better troubleshooting.

There are myriad relational database management systems in combination of coding languages that can be used for the same data-pipeline needs. The most appropriate choices are highly dependent on the data and context being used; some applications excel best at scalability, flexibility, reliability, and other priorities [9]. Although databases are still technically finite in scale, reaching memory limits remains beyond the scope of most scientific labs. For instance, an MS Access database has a memory size limit of 2 GB, which would be a data set on the order of hundreds of thousands to millions of entries depending on the data and number of fields. Most labs will never have experimental needs of this magnitude, but if they did then spreadsheet software would be far beyond their effective limits anyway. In comparison, business-level relational database management systems can handle data sets of larger magnitudes while processing millions of transactions simultaneously [29]. Part of the reason databases are not commonly used in scientific laboratories is that past experiments rarely crest needs of such data magnitudes, so easy-to-use spreadsheet software became widespread instead. A significant investment required to make these methods function, however, is the time needed to plan the data-pipeline and learn SQL for using databases (Figs 3 and 4). Although coding experience greatly hastens the process, most will need to learn SQL from scratch. A wealth of documentation is available online through extensive documentation by developers, as well as free SQL tutorials such as at Codecademy, W3Schools, and SQLBolt [10-12]. Some alternatives that require subscriptions do exist, however, such as the program teaching website Lynda [38]; further reading about database basics can be found online . In an academic setting, good lab buy-in and robust systems can outlast their creators and help facilitate many years of projects across multiple students. This can be accomplished through the creation of guidelines and implementation steps during setup. Indeed, there is high value for all researchers in having a well-functioning joint data-pipeline and database system.

Other benefits of this methodology include the ability to employ automated methods for converting raw data into structured formats, ease of use once stored inside the database, and constant re-updating and re-aggregation of datasets (Fig 3). It is also possible to pull multiple variables' worth of information from a single data file and automate the data-pipeline to do so when prompted. In the context shown, commonly available and economical software, Matlab and MS Access, was used to achieve results demonstrating that expensive and niche software packages are not mandatory in achieving a functional database. Given the limited reach of most laboratories' research funds, the ability to increase the efficiency of database management is a priceless commodity.

In conclusion, as scientific data sets become more complex, databases become increasingly more important for the scientific community and have great potential to be as commonplace as and even more effective than current widespread spreadsheet usage for data storage. Issues with data transparency and replicability in science will only continue to expand in the future as data sets continue to grow in size and complexity, highlighting the importance of more widespread adoption of databases and automated data-pipeline methods for general scientific needs now and into the future.

**ACKNOWLEDGMENTS:**

This work is supported by the National Heart, Lung, and Blood Institute at the National Institutes of Health, grant number R01 HL129008.

The authors especially thank the LMNA gene mutation family members for their participation in the study. We also would like to thank Linda McCarthy for her assistance with cell culture and maintaining the lab spaces, Nasam Chokr for her participation in cell imaging and the nuclei data analysis, and Michael A. Grosberg for his pertinent advice with setting up our initial Microsoft Access database as well as answering other technical questions.

**DISCLOSURES: None**

**REFERENCES:**

1 Cavin, R. K., Lugli, P. & Zhirnov, V. V. Science and engineering beyond Moore's law. *Proceedings of the IEEE.* 100 (Special Centennial Issue), 1720-1749 (2012).

2 Mast, F. D., Ratushny, A. V. & Aitchison, J. D. Systems cell biology. *The Journal of Cell Biology.* 206 (6), 695-706 (2014).

3 Barone, L., Williams, J. & Micklos, D. Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators. *PLoS Computational Biology.* 13 (10), e1005755 (2017).

4 Gandomi, A. & Haider, M. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management.* 35 (2), 137-144 (2015).

5 Siddiqa, A. *et al.* A survey of big data management: Taxonomy and state-of-the-art. *Journal of Network and Computer Applications.* 71 151-166 (2016).

6 Anderson, C. in *Wired Magazine* (2008).

7 Broman, K. W. & Woo, K. H. Data Organization in Spreadsheets. *The American Statistician.* 72 (1), 2-10, doi:10.1080/00031305.2017.1375989, (2018).
8 Lee, H. *et al.* How I do it: a practical database management system to assist clinical research teams with data collection, organization, and reporting. *Acad Radiol.* 22 (4), 527-533 (2015).
9 Bassil, Y. A comparative study on the performance of the Top DBMS systems. *arXiv preprint arXiv:1205.2889.* (2012).
10 *Learn SQL - Codecademy*, <https://www.codecademy.com/learn/learn-sql> (2018).
11 *SQL Tutorial - w3schools.com*, <https://www.w3schools.com/sql/> (2018).
12 *Introduction to SQL - SQLBolt*, <https://sqlbolt.com/> (2018).
13 Pedersen, T. B. & Jensen, C. S. Multidimensional database technology. *Computer.* 34 (12), 40-46 (2001).
14 Győrödi, C., Gyorodi, R. & Sotoc, R. *A Comparative Study of Relational and Non-Relational Database Models in a Web- Based Application*. Vol. 6 (2015).
15 Nayak, A., Poriya, A. & Poojary, D. Type of NOSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems.* 5 (4), 16-19 (2013).
16 Lei, C., Feng, D., Wei, C., Ai-xin, Z. & Zhen-hu, C. The application of multidimensional data analysis in the EIA database of electric industry. *Procedia Environmental Sciences.* 10 1210-1215 (2011).
17 Soranno, P. A. *et al.* Building a multi-scaled geospatial temporal ecology database from disparate data sources: fostering open science and data reuse. *GigaScience.* 4 (1), doi:10.1186/s13742-015-0067-4, (2015).
18 Edwards, P. Questionnaires in clinical trials: guidelines for optimal design and administration. *Trials.* 11 2-2, doi:10.1186/1745-6215-11-2, (2010).
19 Richards, M. A. *et al.* MediaDB: A Database of Microbial Growth Conditions in Defined Media. *PLoS ONE.* 9 (8), e103548, doi:10.1371/journal.pone.0103548, (2014).
20 Core, J. Q. *et al.* Age of heart disease presentation and dysmorphic nuclei in patients with LMNA mutations. *PLoS ONE.* 12 (11), e0188256, doi:10.1371/journal.pone.0188256, (2017).
21 Drew, N. K., Johnsen, N. E., Core, J. Q. & Grosberg, A. Multiscale Characterization of Engineered Cardiac Tissue Architecture. *J Biomech Eng.* 138 (11), 111003-111003-111008, doi:10.1115/1.4034656, (2016).
22 Zaragoza, M. V. *et al.* Exome Sequencing Identifies a Novel LMNA Splice-Site Mutation and Multigenic Heterozygosity of Potential Modifiers in a Family with Sick Sinus Syndrome, Dilated Cardiomyopathy, and Sudden Cardiac Death. *PLoS ONE.* 11 (5), doi:10.1371/journal.pone.0155421, (2016).
23 Zaragoza, M., Nguyen, C., Widyastuti, H., McCarthy, L. & Grosberg, A. Dupuytren's and Ledderhose Diseases in a Family with LMNA-Related Cardiomyopathy and a Novel Variant in the ASTE1 Gene. *Cells.* 6 (4), 40 (2017).
24 Zaragoza, M. V., Hakim, S. A., Hoang, V. & Elliott, A. M. Heart-hand syndrome IV: a second family with LMNA-related cardiomyopathy and brachydactyly. *Clin Genet.* 91 (3), 499-500, doi:10.1111/cge.12870, (2017).
25 Eriksson, M. *et al.* Recurrent de novo point mutations in lamin A cause Hutchinson-

Gilford progeria syndrome. *Nature.* 423 (6937), 293-298, doi:http://www.nature.com/nature/journal/v423/n6937/suppinfo/nature01629_S1.html, (2003).
26 Drew, N. K., Eagleson, M. A., Baldo Jr, D. B., Parker, K. K. & Grosberg, A. Metrics for Assessing Cytoskeletal Orientational Correlations and Consistency. *PLoS Computational Biology.* 11 (4), e1004190, doi:10.1371/journal.pcbi.1004190, (2015).
27 Hamley, I. W. *Introduction to soft matter: synthetic and biological self-assembling materials*. (John Wiley & Sons, 2013).
28 Grosberg, A., Alford, P. W., McCain, M. L. & Parker, K. K. Ensembles of engineered cardiac tissues for physiological and pharmacological study: Heart on a chip. *Lab Chip.* 11 (24), 4165-4173 (2011).
29 Hey, T. T., A. in *Grid Computing: Making the Global Infrastructure a Reality* Ch. 36, (John Wiley & Sons, Ltd, 2003).
30 Wardle, M. & Sadler, M. How to set up a clinical database. *Practical Neurology.* 16 (1), 70-74, doi:10.1136/practneurol-2015-001300, (2016).
31 Kerr, W. T., Lau, E. P., Owens, G. E. & Trefler, A. The future of medical diagnostics: large digitized databases. *The Yale journal of biology and medicine.* 85 (3), 363 (2012).
32 Laulederkind, S. J. *et al.* The Rat Genome Database curation tool suite: a set of optimized software tools enabling efficient acquisition, organization, and presentation of biological data. *Database.* 2011 (2011).
33 Harris, P. A. *et al.* Research electronic data capture (REDCap)--a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of biomedical informatics.* 42 (2), 377-381, doi:10.1016/j.jbi.2008.08.010, (2009).
34 Panko, R. R. What we know about spreadsheet errors. *Journal of Organizational and End User Computing (JOEUC).* 10 (2), 15-21 (1998).
35 Ziemann, M., Eren, Y. & El-Osta, A. Gene name errors are widespread in the scientific literature. *Genome biology.* 17 (1), 177 (2016).
36 (NIH), N. I. o. H. *Rigor and Reproducibility*, <https://grants.nih.gov/reproducibility/index.htm> (2018).
37 Hofseth, L. J. Getting rigorous with scientific rigor. *Carcinogenesis.* 39 (1), 21-25 (2017).
38 *SQL Training and Tutorials - Lynda.com*, <https://www.lynda.com/SQL-training-tutorials/446-0.html> (2018).