Implementing a DNA Database- Relational Algebra with Synthetic DNA

Lee Organick

March 14, 2018

1 Problem to Solve

When tasks are parallelized in silica, they are, ideally, simply divided into smaller chunks for processing so there are many chunks executing one task at a time simultaneously. However, this is often not the case, especially if tasks are dependent on one another, and there is also thrashing to consider. It is well known that distributed computing does not scale simply.

However, chemical processes such as DNA strand interactions occur truly in parallel. All strands in solution (i.e., trillions of strands of DNA in a few microliters of water) interact with each other in all possible ways in fractions of a second. While this property makes modeling DNA strand interactions extremely time-consuming, it is a property we are just now beginning to take advantage of. Until now there has been only the most rudimentary idea that DNA kinetics could be used to speed up large-scale queries by translating bits to nucleotides, but this paper thoroughly explores the idea of implementing a DNA-based database and evaluates its practicality on several different aspects.

2 (Very) Brief Biology Background

DNA is comprised of four different bases called adenine, cytosine, thymine, and guanine which are referred to as "A", "C", "T" and "G" respectively. Simply put, A's and T's will only bind with each other, and G's and C's will only bind with each other.

Of note, G's and C's bind "tighter" to each other than A's and T's. For optimal binding efficiency, a roughly equal proportion of A:T:G:C should be present. In addition, homopolymers (multiple of a single base in a row) should be avoided. Ex. A sequence of $AATTTTCGATCCCCC$ is much less stable than a sequence of similar length where the large T and C homopolymers are not present.

Lastly, DNA can be either single-stranded (ssDNA) or double-stranded (dsDNA). It is convenient to think of dsDNA as the classic double-helix ladder, while ssDNA is missing one half of the ladder lengthwise. Each half of the ladder has a "direction", with one end of the ssDNA strand being 3' and one being 5'. When two ssDNA strands come together, they align so that the 5' end of one strand pairs with the 3' strand of another, so a strand notated

as *5' ATTCG 3'* will happily bind with *3' TAAGC 5'* but will not bind with *5' TAAGC 3'*. This property is vital for the implementation of many of the operations discussed here.

3 Description of Approach

3.1 Methods

For this project, I have examined various relational algebra operations and a few other functions commonly found in DBMS's. The feasibility of performing those operations with DNA is presented below with a brief sketch of a functional schematic for each operation. If the operation can be used, its limitations have been discussed as well.

To further explore the feasibility of DNA Databases, there is in-depth analysis on the notoriously difficult JOIN operation and by extension, semi-joins, anti-joins, and equi-joins by modeling their efficacy using NUPACK, the standard DNA interaction modeling software. Their scalability is also examined.

3.2 Limitations

While NUPACK accurately estimates the amount of crosstalk present between DNA strands, this is only a preliminary tool for several reasons. First, modeling DNA strand interactions is extremely computationally expensive. Every base's interaction with every other base within the same strand and with every other strand in solution must be quantified, and the bases surrounding each base of interest must also be taken into account. To illustrate this, when using a NUPACK python package developed specifically to increase computation speed (better use of internal distributed computing), modeling two strands' interactions with each other took less than a second, four strands took roughly 3 seconds, six strands took a minute, eight strands took roughly half an hour, and ten strands took well over an hour, and twelve strands I stopped after it had run for several hours.

Second, to get data in a timely manner, one must limit NUPACK's simulations to the size of complex you expect to see. For example, with two unique strands NUPACK might find a way for two copies of each strand to bind to each other, forming a four-stranded complex. Here, by necessity, the estimated complex size was limited to two (two strands binding to each other correctly). However, this does not hugely affect these simulations (past experiments have shown this to be negligible though, with quantities of large complexes many orders of magnitude smaller than the small complex formations).

4 Related Work

Synthetic biology is when biological molecules are engineered for purposes far wider than the applications they're organically found in. For example, simple computations of OR, AND, and NOT gates have been done in various labs for over a decade[1] and had been

discussed for years prior. However, large scale circuits and computation with DNA has remained elusive.

Similarly, the idea for storing data in synthetic DNA has been around at least since the 1960's[2], and only recently has been seen as a potentially viable method of archival data storage. DNA is an attractive storage medium because it's orders of magnitude denser than tape (the current commercial method of storing data long-term), and DNA lasts thousands of years longer than tape. Most notably, the feasibility of DNA data storage was demonstrated by UW's own Molecular Information Systems Lab (MISL) in a paper published in Nature Biotechnology in February of 2018[3]. In this paper, in which I played an integral part, we demonstrated that random access could be done on DNA molecules storing over 30 files totaling 200MB of information. While this work is essentially just a simple SELECT function, it demonstrates that large quantities of DNA can be manipulated, individually selected for and recovered with 0-bit error. However, the designs that allowed for high fidelity selection in this project are very similar to the techniques proposed for various JOIN operations in this paper, with the fundamental differences being that I did not screen my designed binding regions for self-complimentarity prior to the presented simulations, and here I have examined how multiple strands would come together to form much larger complexes than prior work (where there are no complexes, only primer binding and replication).

Inspired by logic gates described by various synthetic biologists, and now armed with our large-scale proof of concept that DNA molecules can be selectively manipulated with great success, we began to wonder if DNA could also be used to do certain types of computations faster than traditional computers in silica. Other members of MISL are currently in the process of implementing fuzzy select as a faster way of selecting desired tuples, which they call MASS (molecular accelerated similarity search)[4]. This paper is currently under review and cannot be distributed, but in summary by leveraging the most useful characteristics of both silica and biochemistry they have designed and done preliminary wet lab experiments on a system in which time spent searching over feature vectors is constant. Once the large overhead of synthesizing, and sequencing the retrieved DNA is factored in, they have been able to determine that it would be advantageous to use MASS on datasets with greater than 1,010 feature vectors when compared to disk-based systems performing linear scans over the entirety of a dataset.

Motivated to explore more in the DNA storage space, Microsoft computer architects have also proposed some rudimentary techniques for implementing relational algebra operations on a relational database[5]. However, they rely heavily on CRISPR, a molecular technique with extreme potential but with lots of extra optimization and error incurred. The schematics proposed in this paper rely on much simpler techniques without compromising flexibility. However, if one wanted to edit the database, CRISPR would be the reasonable solution.

Emboldened by recent successes in DNA data storage and with the rise of synthetic biology driving DNA synthesis and sequencing costs down, this is indeed the right time to begin exploring ways in which silica systems could be supplemented with wet-computing.

5 Accomplishments

5.1 Successes

Presented here are ways to implement almost every basic relational algebra function and notes on how to implement a select few functions which are not strictly encapsulated by relational algebra, but which are common in DBMS's. The table below provides a summary of the functions were examined, and whether or not they were feasible and previously documented. Figure 1 below provides a visual aid for understanding how a basic schematic for each operation might look.

Operation	*Feasible?*	*Previously Documented?*
SELECT	Yes	Yes
Fuzzy SELECT	Yes	Yes
Multi-Conditional SELECT	Yes	No
Projection (DISTINCT)	Yes*	No**
RENAME	N/A	N/A
UNION	Yes	No
INTERSECTION	Yes	No
ANTI-JOIN	Yes*	No
NATURAL JOIN	Yes	Yes
SEMI-JOIN	Yes	No
EQUI-JOIN	Yes	No
COUNT	Yes	Yes
LIMIT	Yes*	Yes

Table 1: Operation Summary Table

*Not a true, traditional implementation, but similar enough to be of use
**In genomic DNA, but not in synthetic DNA

5.2 Implementations

For convenience, a visual guide for each operation can be found on page 8.

SELECT - Previously documented [3], random-access (i.e., a single SELECT) is performed using two unique 20 nucleotide probes that selectively bind to their corresponding (reverse complimentary) sequences in the tuple of interest. These sequences exponentially amplify via PCR (a common molecular biology technique) so that they dominate the collection of DNA and when sequenced, are virtually all that remain of the original pool of DNA. It has been shown to work remarkably well in complex pools of DNA [3].

FUZZY SELECT - Previously documented [4], there is a 40bp query attached to magnetic beads which is then introduced to 100bp target sequences. They query bound to target sequences when it was sufficiently similar to make the bind energetically favorable.

Beads were filtered so only bound DNA was extracted. However, while preliminary results show some success, there's still a large percentage of false positives and false negatives associated with the search. More fundamental research of DNA kinetics is needed to make this a viable search method. Luckily, this fundamental exploration is underway and continues to be developed [6].

MULTI-CONDITIONAL SELECT - Just the same as fuzzy select, however the process is repeated for as many conditions as desired. The query attached to magnetic beads releases the products of the search and excess reagents and query materials are washed away, then another query attached to magnetic beads is introduced and the process continues. Unfortunately, the greater the number of queries, the greater the error introduced will be, so considerable tuning would be necessary to ensure adequate fidelity for multiple selections.

PROJECTION - Also known as DISTINCT, this is not a true projection because we cannot limit the results to the same number of copies per unique value. (See *LIMIT* for more on this matter.) However, we *can* return only the attribute(s) of interest separate from the rest of the tuple's data. Tuple strands are designed so that there is a "divider" between each attribute. If only the second attribute is desired (i.e., SELECT DISTINCT attributeTwo), and we vary the length of the attributes such that when the desired attribute(s) is selected with the correct corresponding enzymes (which act exactly like molecular scissors), it can be distinguished by length from the other resulting fragments. Next, filter the sample via gel electrophoresis to separate the fragments by mass (length) so only the fragments of the desired size are selected for recovery. Yet as mentioned above, the copy number of each value in the attribute(s) will vary, but via sequencing we *will* be able to see all values.

RENAME - Not applicable with DNA based databases, as renaming is not a property of biochemistry. However, software being used to interface with these systems should have the ability for the user to rename data as in common DBMS's.

UNION - Assuming the layout of the tables is known beforehand so it is ensured that they are set-operation-compatible, simply mix the samples together. However, once combined it could be impossible to separate the data back out depending on the initial design of the strand. Ideally, easily identified markers (ex. length differences) would allow for simple re-separation of the data.

INTERSECTION - All tables to be analyzed for intersection must follow the same rigid set of rules (i.e., how long each attribute can be, what the encoding scheme is for each value of each attribute, how many attributes there are). Synthesize the two tables to be intersected in DNA, but synthesize one of the pools (AKA tables) to be the reverse compliment (i.e., *5'- Employee name–Age–ID number -3'* for one strand layout, and *5'- ID number–Age–Employee name -3'* for the other so identical tuples will hybridize to form dsDNA and can be selectively filtered by mass (via 2D-SDE, agarose gel, poly-acrylamide gel, etc.).

ANTI-JOIN - Unfortunately it is not possible to do a true ANTI-JOIN, in which results from Table A-Table B would differ from Table B-Table A. However, we can get *all* differences between Table A and B. The setup is exactly the same as INTERSECTION, except when filtering at the last step, the smaller mass is selected (since the ssDNA strands

will be the difference between tables and therefore have half the mass).

NATURAL JOIN - This could be done with virtually any number of tables desired, but you would have to know the number of tables beforehand in order to design optimal join sequences. The number of tuples would have to be carefully monitored, because you will need many more copies of each strand as there are tuples in the table to be joined with it. For example, if you had 1B tuples in one table being joined with a table with 100 tuples, you would need at least 100B copies of each tuple in DNA. Now if you were joining a third table with 10 tuples, you'd need at least 1T copies of each tuple that was produced by the first join. This gets impractical rather quickly due to a fundamental limit of how concentrated DNA can get in practical working volumes before it's a solid mass of DNA and loses it beneficial properties of reacting in solution. In one test tube per table, a "table id" approximately 20-30bp long would be attached to the front of each tuple (via ligation, so the reaction does not depend on the first sequences in the tuple). The reverse compliment of the table id would be attached to the end of each tuple. Thus, when the tables are mixed together and primers to the ids are added, long, assembled dsDNA strands will form where each strand is n tuples long, where n is the number of tables.

SEMI-JOIN - The design of the tables is so constrained by the melting temperatures that this method may only have limited applications (ideally, tables would have minimal attributes to join on, because the more attributes are joined on, the higher the melting temperature will be and at some point, standard PCR [i.e., selection] techniques will not work). Tables need to be designed so the key(s) that would overlap are at one end of the strand. One table will need to be synthesized so that it is the reverse complement of the other. Thus, identical values of tuples from the two tables will overlap and form double stranded parts of DNA with ssDNA overhangs. When a polymerase is introduced to make the overhang double stranded, only one table will be extended to incorporate the other table's data; the other table will remain unchanged.

EQUI-JOIN -See SEMI-JOIN

COUNT - Assuming you are left with only your desired strands to count, this is easily done via DNA quantification. Even without any prior knowledge, a gel could be used to determine the length and whether the sample is dsDNA or ssDNA and with this information, various basic quantification techniques can be used depending on the level of accuracy wanted (ex. nanodrop, qubit and qPCR for increasingly accurate, but more time consuming, quantifications). However, this method will never be as accurate as counting items in silica due to instrument, sample, and prep variation.

LIMIT - In the wet lab, it is currently impossible to simply select 10 molecules of something. Or even exactly 1M. (There's a long standing joke in the field that if you find a way to do this, you will get a Noble Prize.) However, one can easily take a proportion of the returned tuples by easily estimating the count of the returned tuples and then limiting the final returned tuples to any proportion of the original amount desired.

5.3 Limitations

Unfortunately, it is exceptionally computationally expensive to model many strand interactions as described here using the currently existing software (NUPACK). However, adequate simulations can be done to the extent that more variation not modeled is negligible.

Practical wet-lab implementation is also a fundamental concern at this time. With no automation for many of the processes, this is impractical for implementation for many of these functions in part because of cost, but also replication concerns which are often more prevalent with human wet-lab workers. Related, even with the best techniques, there is still variability in wet-lab conditions which make the system very susceptible to perturbations or unexpected variation.

In addition, as mentioned above, at times we still do not understand fundamental DNA kinetics as well as we'd like, so much greater understanding of small scale biochemistry will need to continue being evaluated. Until then, many false-positives and false-negatives will be present in the current systems.

Lastly, for creation of new operations, the process of designing the logistics of how each relational algebra function could be translated into DNA, encoding bits to nucleotides, and then tuning the system and optimizing wet-lab protocols is very time and labor intensive. This was not a trivial task and involves countless sketches and an exceptionally strong grasp of wet lab DNA manipulation techniques in addition to knowledge of what database users want from such a system.

6 Evaluation

6.1 Expected Error Rate

As mentioned above, while reaction kinetics are still not completely understood, enough is known to design "selection footholds" (otherwise known as primer regions) that act with good fidelity, and tools exist to help model these interactions with high enough fidelity to be useful.

Using NUPACK to model these DNA interactions, by setting the maximum complex size to two (see above for more discussion) the following are time-intensive simulations of the percent of final product that forms the correct expected product (see Figure 2 on page 9). It is estimated with high accuracy that for every additional table joined, there is roughly four percent less desired product present. However, this is using a 20 base primer design scheme unique to this project, as I cannot use the scheme published in Organick et al. because of Microsoft's proprietary code for this. I have reason to suspect that using the scheme developed there, the variation would decrease drastically, and the slope would be much less than it is here.

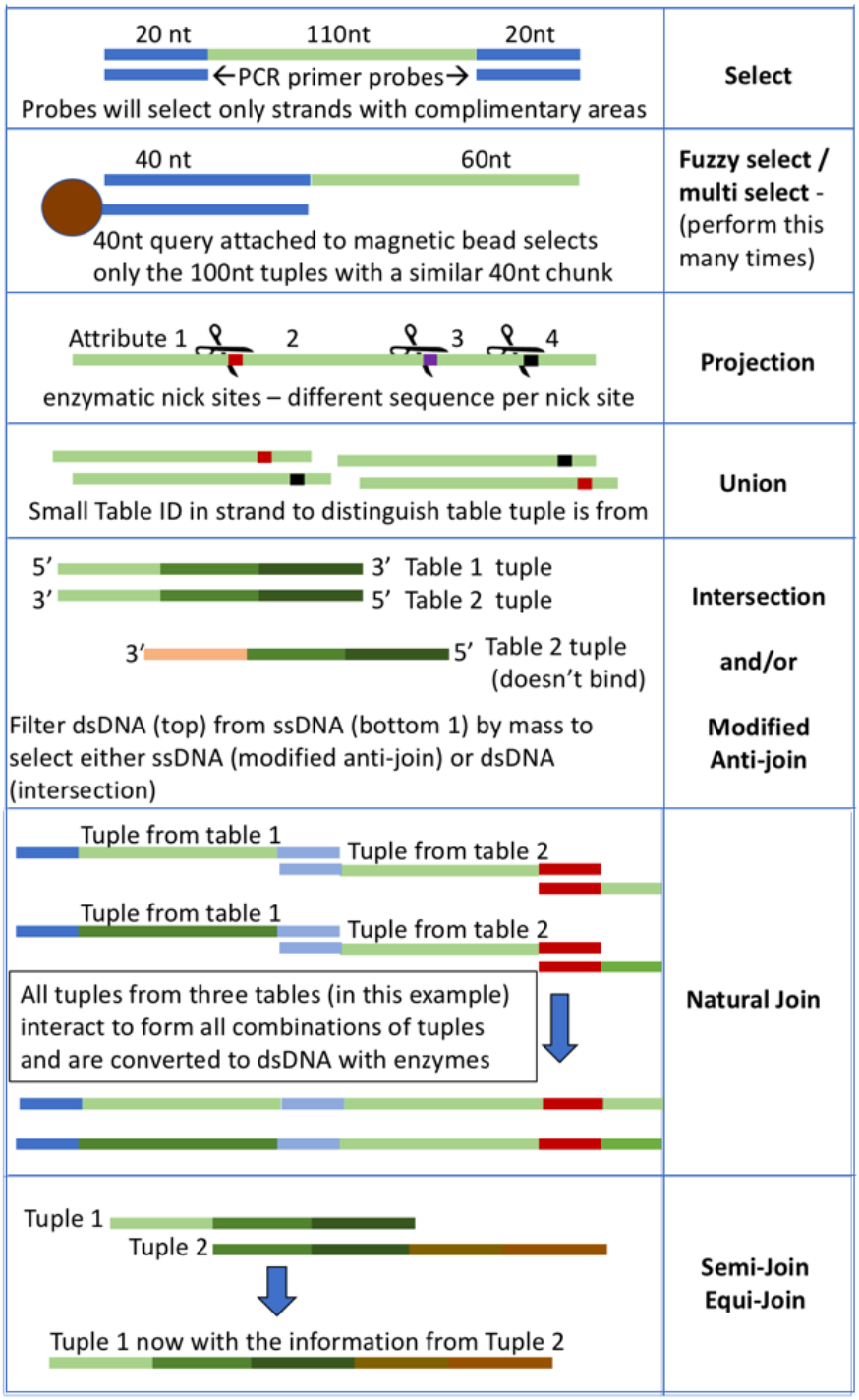

Figure 1: Brief sketch of each implementation

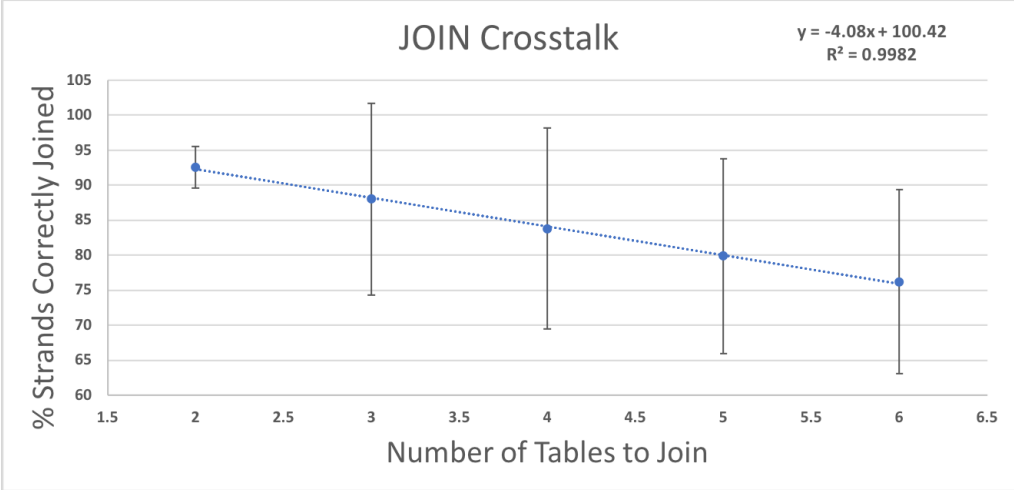

Figure 2: Percent of desired product returned after subsequent JOINS

6.2 Scaling to Large Databases

One concern with having large numbers of molecules interact with each other is that there will be so much cross talk the returned strands will be nonsense. However, much work has been done[3] to ensure large pools of DNA with unique selection footholds with high fidelity. Shown in Figure 3 is the model showing expected pools with many thousands of unique regions long, and many more are possible if the extremely stringent constraints are relaxed, though that would increase the amount of crosstalk.

The next concern is even having enough DNA to perform these functions. Fortunately, as shown in Figure 4, current synthesizers can already handle making large amounts of unique strands with high accuracy[3] and this capability is only expected to increase in the coming years as synthetic biology continues to grow as a field. And if more DNA is wanted, but no more unique strands, simple molecular processes (PCR) enable quick, virtually cost-free exponential amplification.

6.3 Computation Time in Silica vs. DNA

Finally, the point in which it is faster to perform computations in DNA faster than in silica must be explored. See figure 5. For the sake of brevity, only the natural join function is explored in depth here. However, similar relationships are present with other join functions. Here, it is important to note that the time it takes to perform natural join in DNA is linear, but the slope is so gradual more basic research is needed to determine its slope. It is also estimated here (based in experience) that DNA synthesis takes approximately 15 hrs, and the join process itself is on the order of an hour. This is the crucial advantage of computation with DNA because performing natural joins in silica increases in power function time, which despite its initial speed becomes crippling with extremely large datasets.

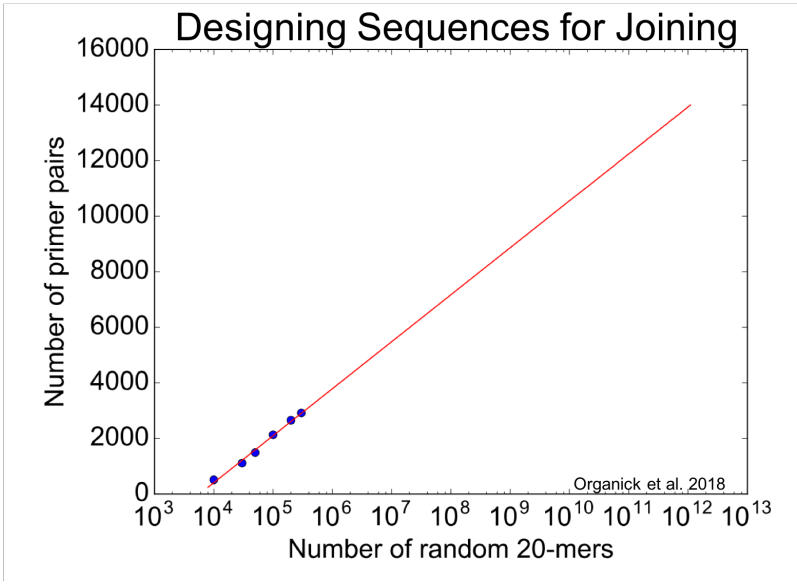

Figure 3: Reprinted from [3] showing the number of 20bp sequences that can exist together in a large pool of DNA with negligible cross-talk with each other using the design scheme in [3].

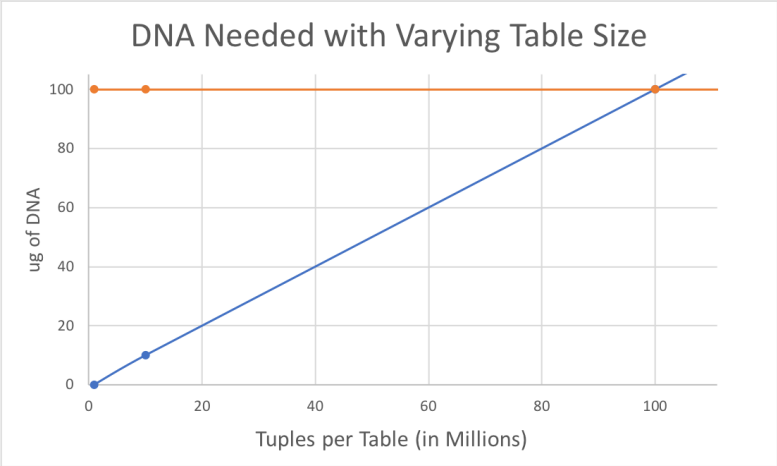

Figure 4: The number of unique strands needed increases the more tuples are present (assuming only two tables in this model), and current technology can synthesize the orange flat line in parallel fairly comfortably.

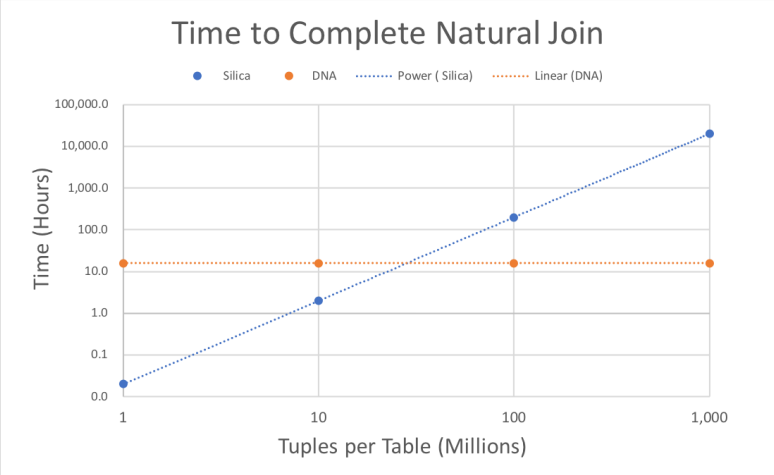

Figure 5: The estimated time needed to perform a natural join assuming it takes 1.2 min to join two tables with 1M tuples each, versus the time it takes to do the exact same computation in synthetic DNA including the overhead of printing the DNA.

7 Conclusions and Future Vision

When dealing with large data sets containing many millions of tuples, at minimum the natural join is indeed faster when implemented in DNA, and there is reason to suspect the same is true for other functions [4]. I plan on continuing to explore these crossover points for other functions in my future graduate work. In addition, crosstalk and wet-lab variability will always be present. Future designs will either have to tolerate false negatives, false positives, and more than likely both at once. Yet with some error correction, these errors will be mitigated perhaps to the point of negligence. However, to make DNA databases a feasible option for future large-scale queries, several key things will need to happen.

First, to mitigate the large time and monetary overhead of synthesizing large datasets in DNA, a flexible, multi-use scheme will have to be implemented. It is usually not practical to perform one or two queries limited to a prior schema, then re-synthesize the entire dataset. Luckily, some molecular processes can help with this (ligation to add desired sequences to the ends of strands and CRISPR to edit sequences within strands are two powerful tools that come to mind).

Second, wet lab operations will never be as accurate as silica systems are without extensive overhead that may slow down the wet lab operations to the point of worthlessness or prohibitively high cost. Implementations of any wet lab process should take this into account and be willing to accept this trade off of speed and accuracy.

And last, automation must make its way into the wet lab to be able to scale this whole process up. Ideally, the ultimate scenario would be for a programmer to write a query, the optimizer for the DBMS would then determine if it was faster to perform the query in silica or in DNA, then automatically encode the data, synthesize, query in the wet lab, read out

the data and return the results some hours (or more likely days) later depending on the amount of data to be encoded and synthesized. Luckily there is precedence for some of this automation pipeline with companies like Emerald Cloud Lab and Transcriptic which all allow scientists to remotely code wet lab procedures to be done in wet lab locations located elsewhere.

However, despite all the work to be done making this a viable part of future optimization, this work indicates that we're already able to perform certain computations faster in DNA than in silica with the limited tools we currently have. It's just going to cost a few tens of thousands of dollars to do it in the near future before synthetic DNA prices drop.

8 References

1. Seelig, Georg, et al. "Enzyme-Free Nucleic Acid Logic Circuits." Science, 8 Dec. 2006, science.sciencemag.org/content/314/5805/1585.

2. Neiman, M.S. "On the molecular memory systems and the directed mutations." Radiotekhnika 1-8, 1965.

3. Organick, Lee, et al. "Random Access in Large-Scale DNA Storage." Nature Biotechnology, 19 Feb. 2018, https://www.nature.com/articles/nbt.4079.

4. Stewart K., Willsey M., Strauss K., Ceze L. "Molecular Accelerators for Data-Intensive Workloads." Under review. 2018.

5. Strauss et al. U.S. Relational DNA Operations. Patent Application 20160371434 filed June 16, 2015.

6. Zhang, Jinny et al. "Predicting DNA Hybridization Kinetics from Sequence." Nature Chemistry, 6 Nov. 2017, https://www.nature.com/articles/nchem.2877