An amateur analysis of the structure of the human genome

John P. Costella
770 5th St NW Apt 1207, Washington, DC 20001-2673, United States

(September 10, 2025)

Abstract

I analyze the raw commercial high-depth whole genome sequencing data for two unrelated humans, one male and one female. I find that their haploid genome is twice as long as the reference human genome, with half of their sequences missing from the reference genome. I speculate that this is true for all humans.

Dedicated to the memory of Charlie Kirk.

Written before today's tragic news. Any joy from it has been destroyed.

1. Introduction and motivation

The plummeting cost of high-depth whole genome sequencing means that amateurs like me can play with the raw data. In the pursuit of a different personal hobby project [1], in September 2024 my wife Sally and I each purchased 100× Whole Genome Sequencing from Nebula Genomics. We received the data about six weeks later. I ultimately ramped up that project in June 2025.

Recently, as a diversion, I wandered from the goal of that project and slightly modified the method I was using so that I could play with and better understand the raw sequencing data itself. The results that I got the next day, August 18, were so surprising and baffling that within 24 hours I started writing up this paper, even though I had no idea what the explanation was. The other project was relegated to the backburner, where it remains to this day. I continued to write up this paper simultaneously with my exploration of the data, and am deciding to leave it in the "near present" tense, with some of my original comments left intact, rather than recasting it into the past tense, as I believe that my fascinating adventure will be more interesting as a warts-and-all narrative than if I were to follow the traditional academic path of editing, polishing, and buffing my work to hide all of the missteps whose random walk ultimately led me to what I believe to be the correct answers.

In Secs. 2 through 5 below I pretend that I know absolutely nothing about the structure of the human genome (not actually too far from the truth), and perform all of my analysis using just the Nebula Genomics raw genomic sequencing data. Remarkably, I find that it's relatively simple to show, just from these two sequencing datasets, that our genetic material must come in chunks of around 6.3 billion bases. But I drop the pretence of ignorance in Sec. 6 when I realize that the results have a fundamental flaw: far too many sequences appear only once in that chunk of 6.3 billion bases than what we know to be the case. In Secs. 7 and 8 I run through two theories for what I did wrong, both of which I then shoot down. At a loss for what to do next, in Sec. 9 I completely break my intention of doing a *de novo* analysis and "cheat from the back of the book" by using the reference human genome [2] to confirm what seems to be a crazy conclusion: the human genome is actually twice as long as we think it is.

In Sec. 10 I confess something omitted from the earlier sections: my early attempt at a more accurate modeling of the distribution of the raw data had failed miserably, which had compounded my initial confusion. Unexcitedly, it was my ignorance of how genomic sequencing machinery actually works that was to blame.

The fog was lifted, but I realized that to claim that we just somehow "misplaced" half of the human genome would be a difficult proposition to swallow just on the basis of statistical arguments. Rather, the "show me the money" move [3] would be to perform a *de novo* assembly of the genome from just our data.

To that end, in Sec. 11 I construct a rudimentary algorithm for growing the sequences from the raw data, one base at a time. Secs. 12 and 13 describe the results of applying it to our data.

Sally and I have decided to make these partial assemblies publicly available on my website [4] for scrutiny. We will make our raw Nebula Genomics data available as well, if someone can host that 0.82 TB of data for us. I also supply all of the code that I used to perform my analyses.

2. The raw data

Sally's two paired-end FASTQ files from Nebula Genomics each contain 1,560,558,378 reads of 150 bases, yielding a total of 468,167,513,400 bases sequenced. Each of mine contain 1,424,404,286 reads of 150 bases, for a total of 427,321,285,800 bases sequenced.

For the other project [1] I had already preprocessed the pair of files for each of us into a single file in a custom format. For both projects I ignore the paired-end nature of the R1 and R2 files, and simply consider them to be two sources of raw data for each of us.

3. Analytical method

I here modify slightly the methodology that I am using for the other project [1] by analyzing the frequencies of every "hexadecigram" (*i.e.*, 16-gram) of 16 consecutive bases in the raw sequencing data. I call the 32-base equivalents "needles" in my other project, since I was searching for a particular needle in the "haystack" of the human genome (spoiler alert: it's not there). Since it's less of a mouthful and easier to type, I call them "needles" rather than "hexadecigrams" for most of this project too, except where the n-gram nature of them is worth emphasizing.

Since the four bases `A`, `C`, `G`, and `T` can be represented by the four dibits (two-bit integers), each 16-base needle in this project can be represented by a 32-bit unsigned integer.

I consider only needles that fall completely within a 150-base read, and I do not include any that straddle an unknown `N` base code in the raw data (infrequent, but present). I ignore the quality data present in the raw data files, and simply take the raw bases at face value, with the understanding that they will have some rate of error. With these rules there are 421,298,960,365 needles for Sally and 384,534,620,798 for me. These numbers are just over 10% smaller than the number of bases in the raw sequence files because I use up the first 15 bases of each 150-base sequence to fill up all but the last position in a 16-base "needle photocopier." I then continue to feed the remaining 135 bases of the 150-base sequence into the photocopier, one base at a time. Each base fed in causes the oldest base to poop out the other end, and the photocopier

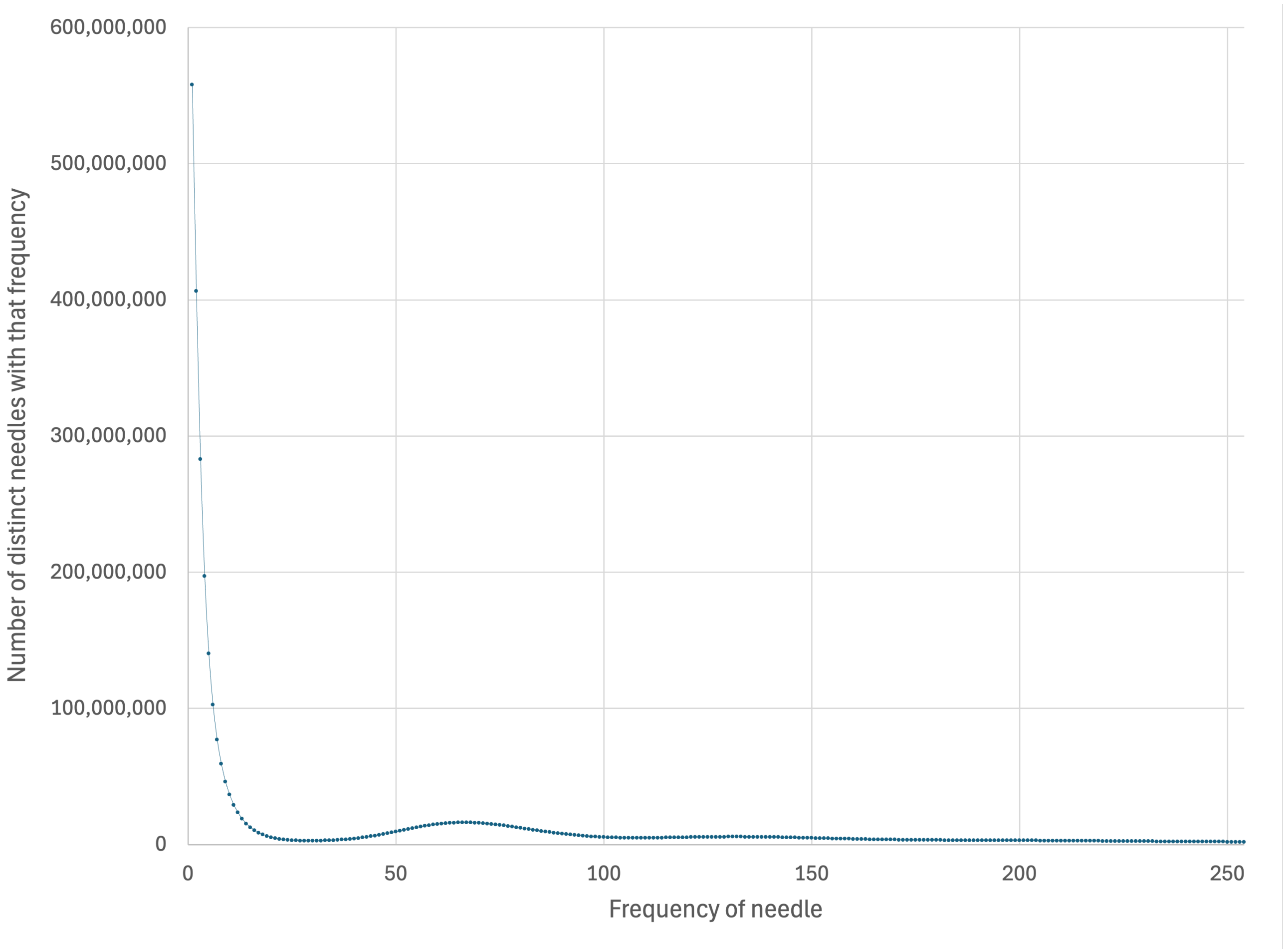

Figure 1: The number of distinct needles for each frequency less than 255 in Sally's data.

prints out another 16-base needle into the output tray for me to collect and subsequently analyze. (It's slightly more than a 10% loss because of those infrequent `N` codes that break the 150-base sequences.)

I then compute the frequencies of all 2^{32} needles for each of our files. Since the maximum frequency for any needle turns out to be 161,029,799 for Sally and 150,161,697 for me, the frequencies can also be represented by 32-bit integers. This allows me to store an entire frequency table in a simple C array of 2^{32} 32-bit integers, which is only 16 GiB in size, and hence easily fits into memory on my laptop. For speed, I save these 16 GiB arrays to storage without any compression. All further processing is done on these arrays.

4. Frequencies of the frequencies

I know from my day job that a very useful way to understand a large frequency table is to analyze the frequency table *of* the frequencies, so I wrote a program to do just that. The results for Sally, for small frequencies, are shown in Fig. 1. We can see some sort of divergence as we approach zero frequency, and some sort of bump around 65. Let's ignore the former for the moment. I rescale the vertical axis in Fig. 2 so that we can take a better look at that bump. A detailed inspection shows that it peaks at a frequency of about 66.5. We now also see that there

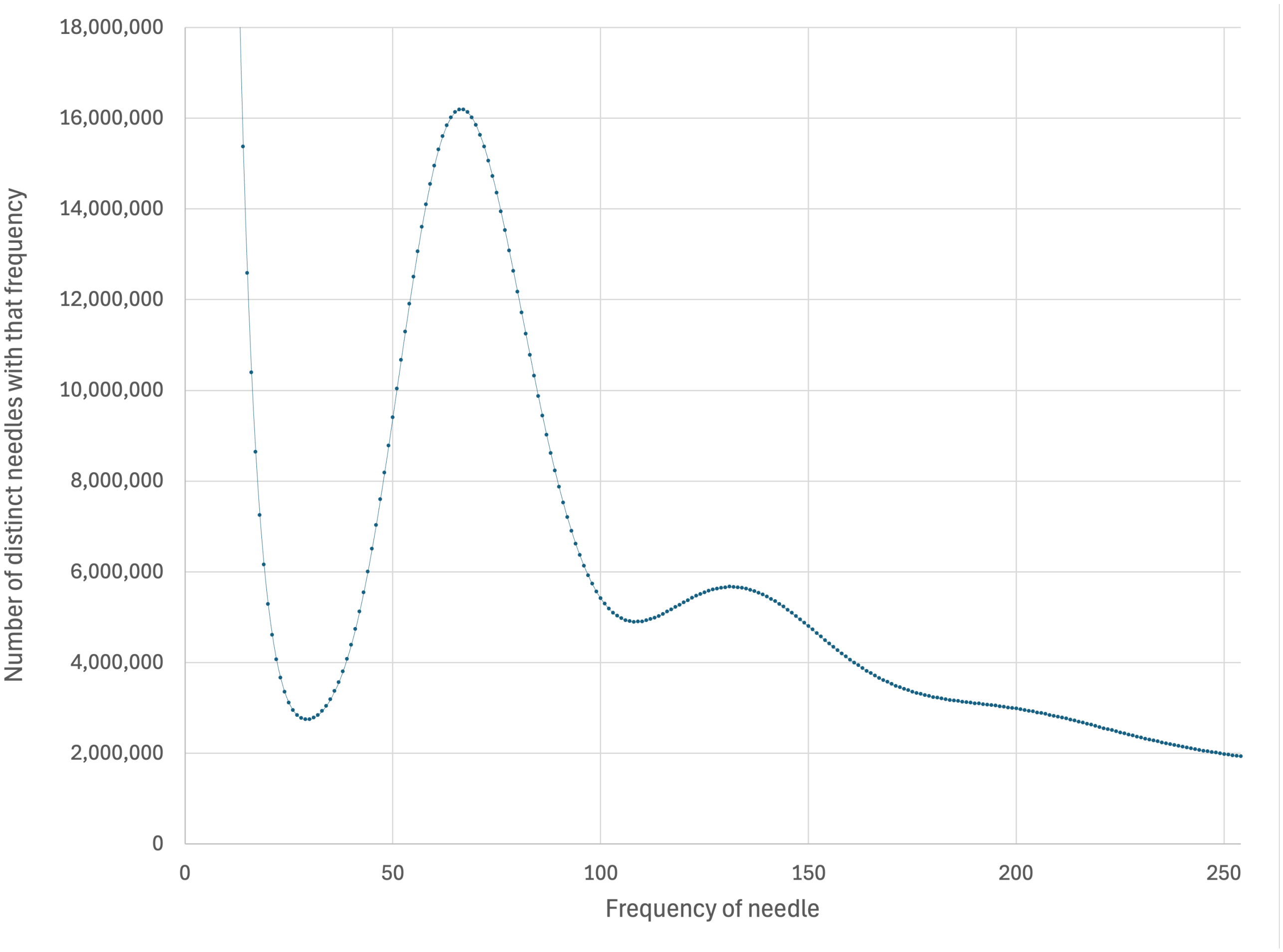

Figure 2: The same graph as Fig. 1, but with the vertical axis rescaled.

is a smaller bump that peaks around 130, with maybe a third one somewhere near 200.

The same graph for my data is shown in Fig. 3. In this case, if we fit the top of the first peak with a parabola, we find that its maximum is at a frequency of about 61.3. The other two bumps are likewise at slightly lower frequencies than Sally's. The ratio of my modal (excluding the low-frequency divergence) frequency of 61.3 to Sally's 66.5 (92.2%) is almost the same as the ratio of my total number of needles to hers (91.2%), which makes intuitive sense.

5. The Millikan oil drop experiment

Robert Millikan's 1909 oil drop experiment [5, 6] is one of the greatest experiments in the history of physics. It's also so simple that even an undergraduate can perform it. I did it myself as an undergraduate. To see with your own eyes the effects of the fundamental quantum of electricity—and to measure its charge—is an experience that every true geek should undertake at least once in their lifetime.

Figs. 2 and 3 remind me of that oil drop experiment. Even if you know nothing about the human genome, you can see with your own eyes that there are peaks for one, two, and maybe even three quanta of . . . something. It seems that this raw genetic material actually only comes in packets of a fixed size, like buying something in bulk from Amazon or Costco that is only available in one size. (Yes, Sally, don't remind me of that large box of 80 disposable coffee cups, when I only wanted a dozen.) A better analogy is that it is all printed in a book, and we have many copies of the same book. The first peak in Figs. 2 and 3 corresponds to words that appear exactly once in the book. I'm going to call these singularly exceptional words "single needles," or just "singles," for convenience. Likewise, the second peak corresponds to those needles that appear exactly twice in the book ("doubles"), and so on. The divergence towards zero probably represents occasional misprints (sequencing errors or contaminants). Each of the curves are spread out because we've torn open boxes full of the books, broken the spine of every book, ripped out all their pages, and run all those pages through a shredder. (I'm now getting visions of Lee Harvey Oswald going on a different sort of rampage in the Texas School Book Depository.) There's nothing left of the original structure of each book, but by grabbing random bits of shredded paper and reading them we'll occasionally see these rare needles. Mathematically, we would expect this random sampling to be described by a Poisson distribution.

Even without that level of mathematical sophistication, we can estimate how many needles there are in each book. Sally has about 421 billion needles in total, and her singles peak is at a frequency of about 66.5, so, on average, one in every 6.33 billion needles or so is a single. I have about 385 billion needles in total and my singles peak is at about 61.3, so one in every 6.28 billion needles or so is a single. If we make the assumption that the actual genetic material is made up of sequences that are far longer than these little 150-base ones that Nebula gave us, then the number of needles will be practically the same as the number of bases (*i.e.*, we won't lose that 10%, but only a far smaller percentage). So our genetic material comes in a book that has about 6.3 billion bases in it.

I find it remarkable that these tiny sequences can—if we have enough of them—tell us something about the structure of the entire human genome. Maybe that's well-known in genomics. As an amateur, it's awe-inspiring.

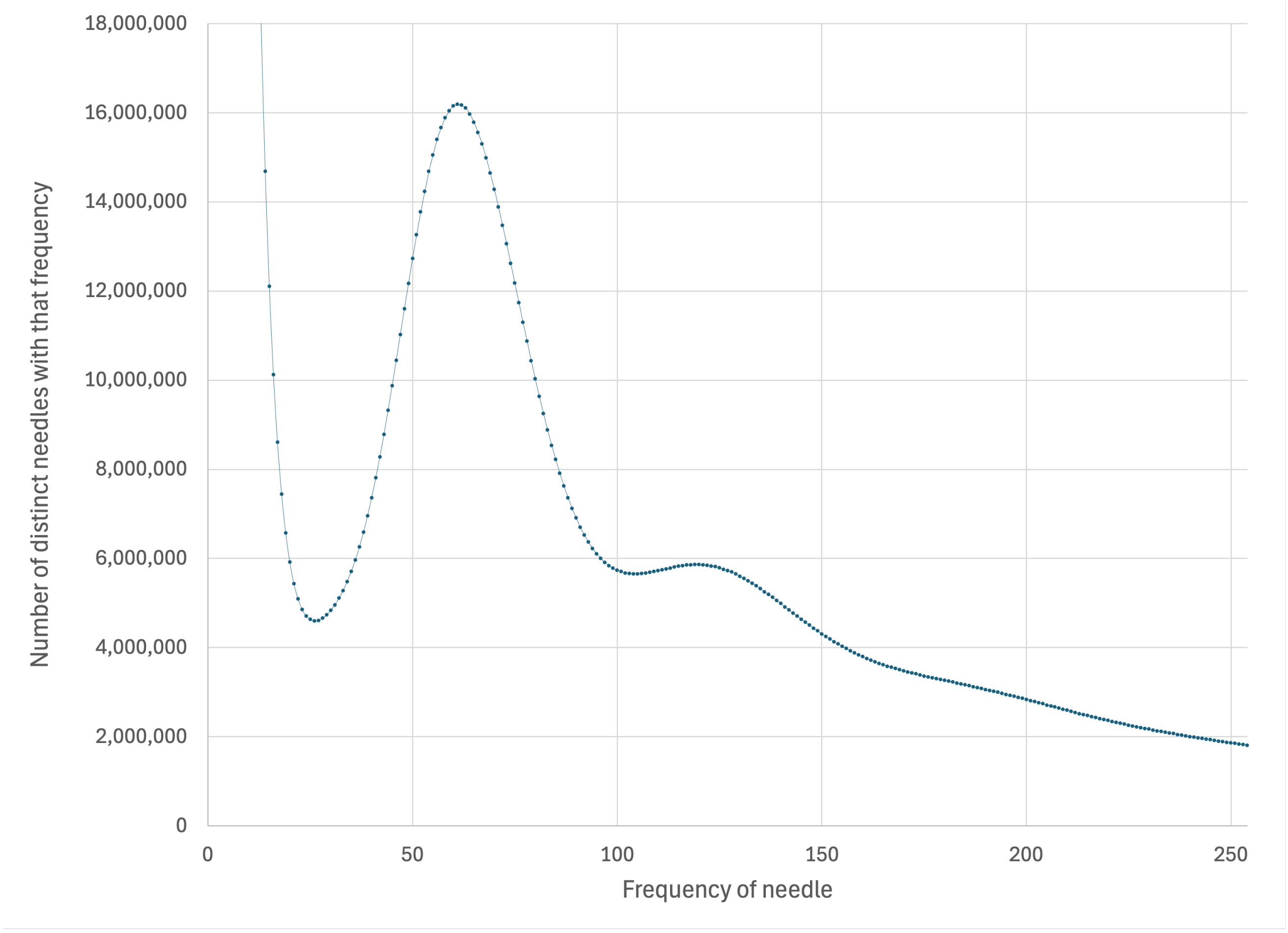

Figure 3: The same as Fig. 2, but for my data.

1 \ 2	A	C	G	T
A	10.07%	5.03%	6.92%	7.73%
C	7.26%	5.14%	1.19%	6.82%
G	6.05%	4.30%	5.29%	4.92%
T	6.37%	5.94%	7.15%	9.83%

(a) Sally

1 \ 2	A	C	G	T
A	10.11%	5.04%	6.91%	7.79%
C	7.26%	5.11%	1.17%	6.81%
G	6.06%	4.27%	5.23%	4.91%
T	6.43%	5.94%	7.14%	9.83%

(b) Me

1 \ 2	A	C	G	T
A	9.71%	5.05%	7.01%	7.66%
C	7.26%	5.20%	1.02%	7.00%
G	6.00%	4.28%	5.24%	5.06%
T	6.46%	5.96%	7.29%	9.80%

(c) Reference genome

Table 1: Bigram frequencies computed directly from the frequency table files.

6. Houston, we have a problem

It didn't take me long to shake off both my awe and my pretense of ignorance to realize that there's something seriously wrong with Figs. 2 and 3: *there are far too many singles.* Even without more sophisticated modeling, if we simply add up the total number of needles in Sally's Fig. 2 between frequencies of, say, 44 and 97, we get an answer of about 43 billion needles. That's 10% of the total number of needles! Likewise, if we do the same for my Fig. 3, for frequencies between, say, 40 and 88, we get about 38 billion needles, which again is around 10%.

That makes no sense, because we know that each haploid genome has 3.1 to 3.2 billion bases; it's only the cell nucleus diploid *pair* of them that has 6.3 to 6.4 billion [7]. So we're seeing 10% of the needles appearing just once in the *diploid* genome. That means that they only appear on one of the two haploid genomes. But we know that the two haploid genomes do not have differences that could add up to 10%. Men have one X and one Y chromosome, of course, which in general are completely different, but they make up only about 3.5% of the total diploid genome. In any case, that's not relevant for Sally's data. The differences between Sally's two haploid genomes, at the scale of these 16-base sequences, should be no more than about 0.1%.

I knew that I had made some sort of mistake.

7. Am I reading half the data backwards?

My first theory for my error was that I somehow misunderstood the format of Nebula's paired-end R1 and R2 FASTQ files, and had read the R2 file backwards. To check this hypothesis I calculated the bigram frequencies from each frequency table file: if I had reversed half the data, the results would be very close to symmetrical around the main diagonal, whereas we know that the actual values are far from symmetrical [8].

The results are shown in Tables 1(a), 1(b), and 1(c), where I also show the results of running the same program against the reference human genome, whose frequency table I compute using a slightly different algorithm directly from the raw data: for ambiguous base codes I include all possible needles, but only allowing up to two consecutive N base codes (because long strings of them are used in the FASTA data file for unknown or uncharacterized parts of the genome).

Clearly, the bigram frequencies are correct. My first theory went down in flames.

8. Are they single-nucleotide variants?

My second theory was that the result *might* make sense if the singles mainly correspond to single-nucleotide variants. My logic was that a single mutated base on one of the two haploid genomes actually affects 16 consecutive needles straddling that base. If all of those 16 mutated needles happen to be ones that don't appear at all in the unmutated haploid genome, then one mutation could lead to up to 16 single needles. In addition, if all of the 16 needles in the diploid genome originally appeared just once in each haploid genome, then the mutation would leave all of the unmutated needles as singles as well. Putting these together, one mutation could actually create up to 32 singles; or, conversely, the number of mutations could actually be up to a factor of 32 smaller than the percentage of singles in the data—namely, about 0.3%. This is still three times too high, and even this low limit would rely on most of those 32 singles actually being produced from the mutation. I don't know how likely that would be, but at least this theory pushes the numbers in the right direction.

To test this theory I wrote a program to compute the joint frequency table of the frequency of any given needle in Sally's data against its frequency in mine, focusing on the square domain where both frequencies are less than 255. My expectation was that the singles peak of Sally's data would be *split* in my data into peaks around 0, 61.3, and 122.6: around 61.3 for those singles caused by mutations that Sally and I both have, and around 0 and 122.6 for mutations that she has but I don't. Likewise, the singles peak of my data would be split into three peaks for her data, around 0, 66.5, and 133. In other words, I expected the peak at $(66.5, 61.3)$ to be split in both directions, with peaks at the five points that are the extremes and the center of a "+" shape, with the five peaks being at $(66.5, 122.6)$, $(0, 61.3)$, $(66.5, 61.3)$, $(133, 61.3)$, and $(66.5, 0)$.

The results are shown in the density plot of Fig. 4, where again I truncate the height axis to best show the singles peak.

The expected splitting is not there. My second theory also went down in flames.

9. The fog lifts

Some days after shooting down this second theory, I realized that it didn't even make any sense to begin with. If that 10% of needles that are singles correspond to mutations of sequences that in unmutated form appear on both haploid versions of the diploid genome, then *where are the rest of the unmutated needles*? They should represent a much bigger peak around a frequency of about 133 for Sally, and around 122.6 for me. But those peaks in Figs. 2 and 3, or equivalently the peak at $(133, 122.6)$ for the joint distribution in Fig. 4, are absolutely pathetic compared to the singles. So where has our diploid genome gone?

We can confirm that this is a real problem from yet another direction. Nebula Genomics told me by email that the coverage of Sally's data is about 139, *i.e.*, they estimate that every position on the haploid genome is sampled on average 139 times (this is reduced by about 5% from what you get from the raw numbers due to them discounting it for things like duplicate reads). For mine they estimate it to be about 127. Both of those numbers exceed the advertised 100× that

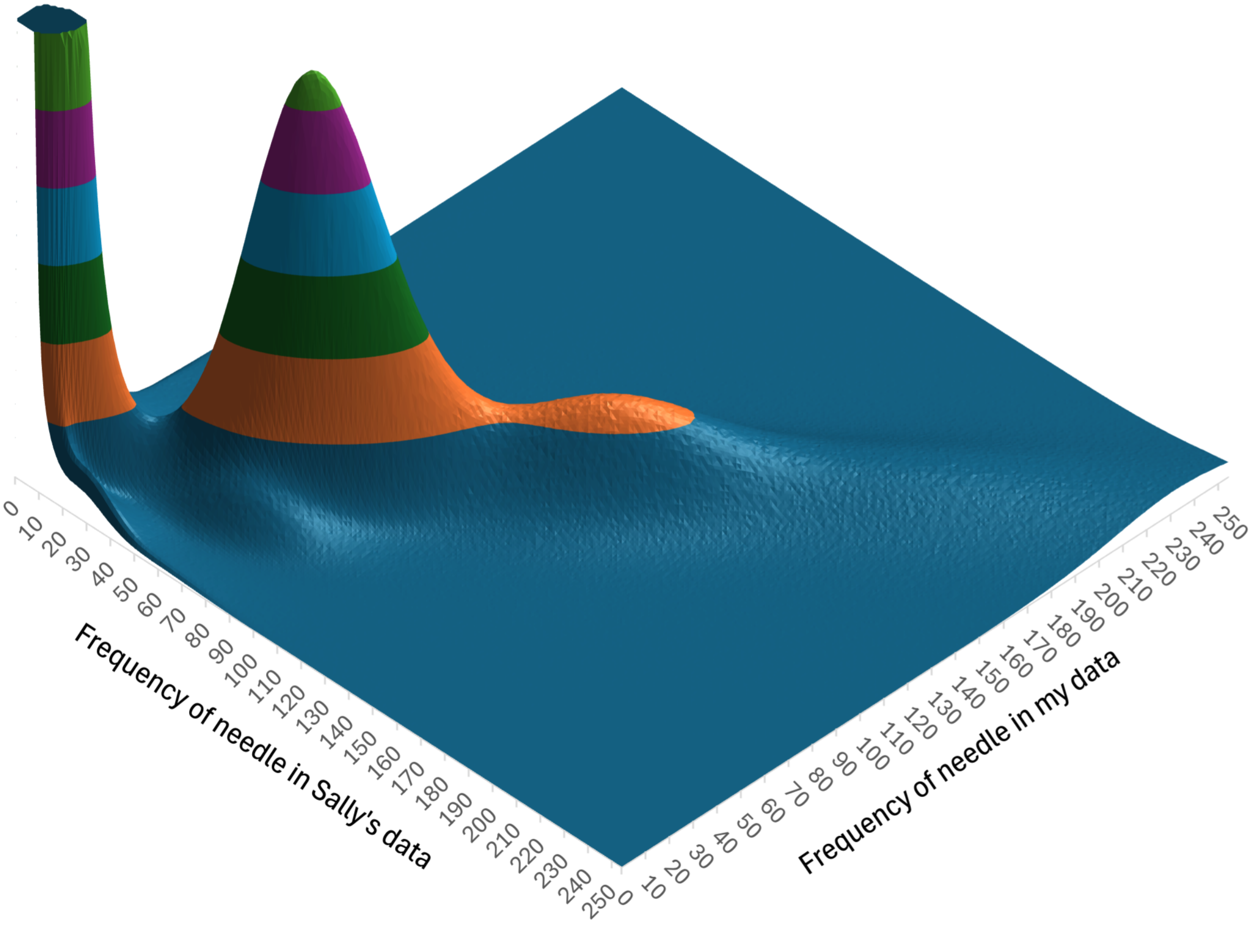

Figure 4: The joint frequency distribution of needles between Sally's data and my data.

we paid for, so we are happy customers. But we can see with our own eyes that the large peak of singles only actually has a coverage of about 66.5 for Sally, and 61.3 for me. Nebula gave us more than what we paid for, but it didn't actually provide the coverage that any of us expected.

It almost feels like *the genome is actually twice as long as we know it to be.* That book of 6.3 billion bases seems to be the *haploid* genome, *not* the diploid one. That would make the diploid genome around *12.6 billion* bases long.

Crazy talk. But we can do a sanity check. If the diploid genome really *did* have 12.6 billion bases in it, then we *would* expect to see *some* needles that appeared only once in Sally's diploid genome—namely, those very ones due to mutation that I described above. The difference now—if the diploid genome really *is* twice as long as it is supposed to be—is that these peaks now would occur at frequencies that are all *half* the values that I described above, namely, $(33.3, 61.3)$, $(0, 30.7)$, $(33.3, 30.7)$, $(66.5, 30.7)$, and $(33.3, 0)$.

You can see the second-last of these in Fig. *4*, where the small bump of a "growth beneath the skin" catches the (artificial) light of this surface plot rendering. There isn't enough data there to say anything else definitive about it, but it's in the right place.

I then thought of another way to check that I'm not "dreamin'" [9]: simply redo the joint distribution in Fig. 4, but filter to *only include those needles contained in the reference human genome.* The results are shown in Fig. 5, which has the same vertical axis scale as Fig. 4. Note that almost all of the divergence near zero has disappeared, which tends to confirm my speculation that it was caused by sequencing errors or contamination. But more importantly, *the singles peak is only half as high as it was in* Fig. *4*. We can confirm this by inverting the filter logic, and only including needles that *do not* appear in the reference genome. This is shown in Fig. 6, which again has the same vertical axis scale as Fig. 4. If you flip between Figs. 5 and 6 you can see that the singles peak is essentially identical in each; in other words, half of the single needles that Sally and I share appear in the reference genome, *but half do not.*

That the peak for doubles is highly suppressed in Fig. 6 is just an artifact of the filtering logic, which puts those needles into Fig. 5 unless *both* appearances in our haploid genome do not appear in the reference genome. We can avoid this bias by instead going back to Fig. 4, but now computing the frequency table for needles between Sally's data and the *reference* genome. The results are shown in Fig. 7. We see that there are slightly more of Sally's singles that don't appear in the reference genome at all than appear once. The doubles in Sally's genome appear on average once in the reference genome. The equivalent joint distribution for my data against the reference genome is shown in Fig. 8, which is very similar. We thus find that not only are half of the singles appearing in each of our genomes completely missing from the reference genome, *but so too are half of our doubles.*

Gaining confidence in my crazy conclusion, I looked back at Fig. 5, where filtering to only those needles contained in the reference genome did such a great job of eclipsing the blinding noise divergence at the origin that I wondered—docked English Cocker Spaniel tail quivering in hope: *could it have actually made my predicted peaks at* $(0, 30.7)$ *and* $(33.3, 0)$ *visible?* A peak of needles appearing only once every 12 billion bases would put the final nail in the coffin of the idea that the haploid genome could only be 3 billion bases long. It's just not possible.

Zooming back into Fig. 5 from the left, I saw Fig. 9.

I had my Eddington experiment.

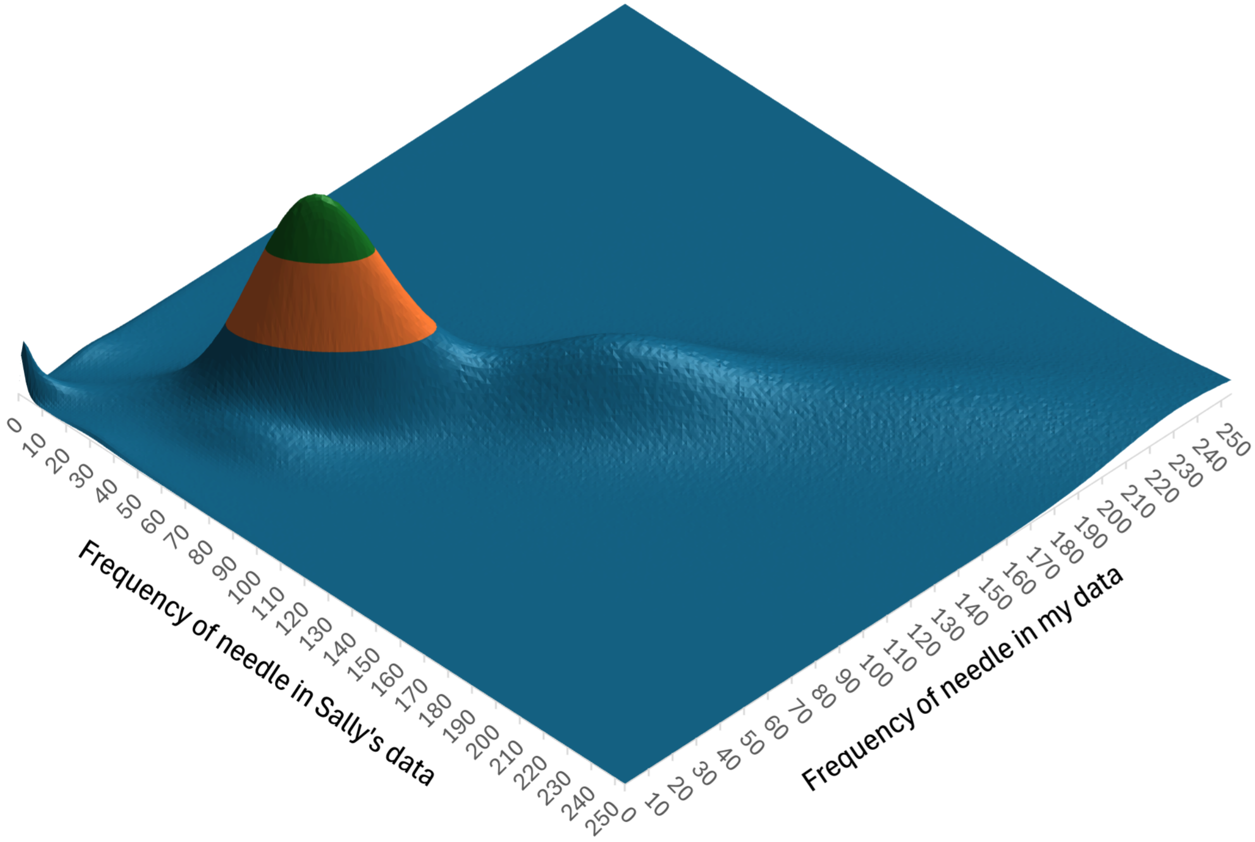

Figure 5: The same as Fig. 4 but only for needles that appear in the reference genome.

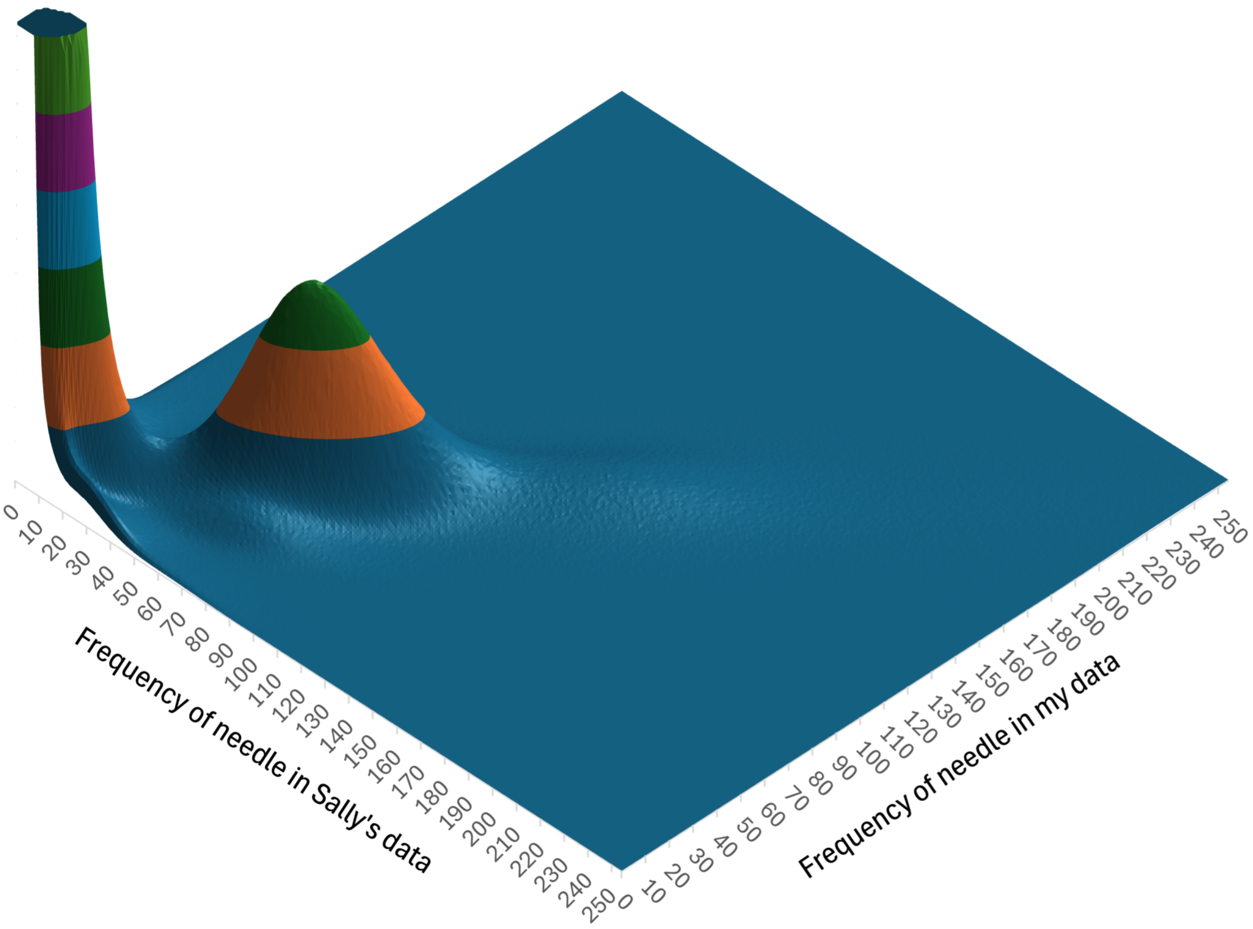

Figure 6: The same as Fig. 4 but for needles that do *not* appear in the reference genome.

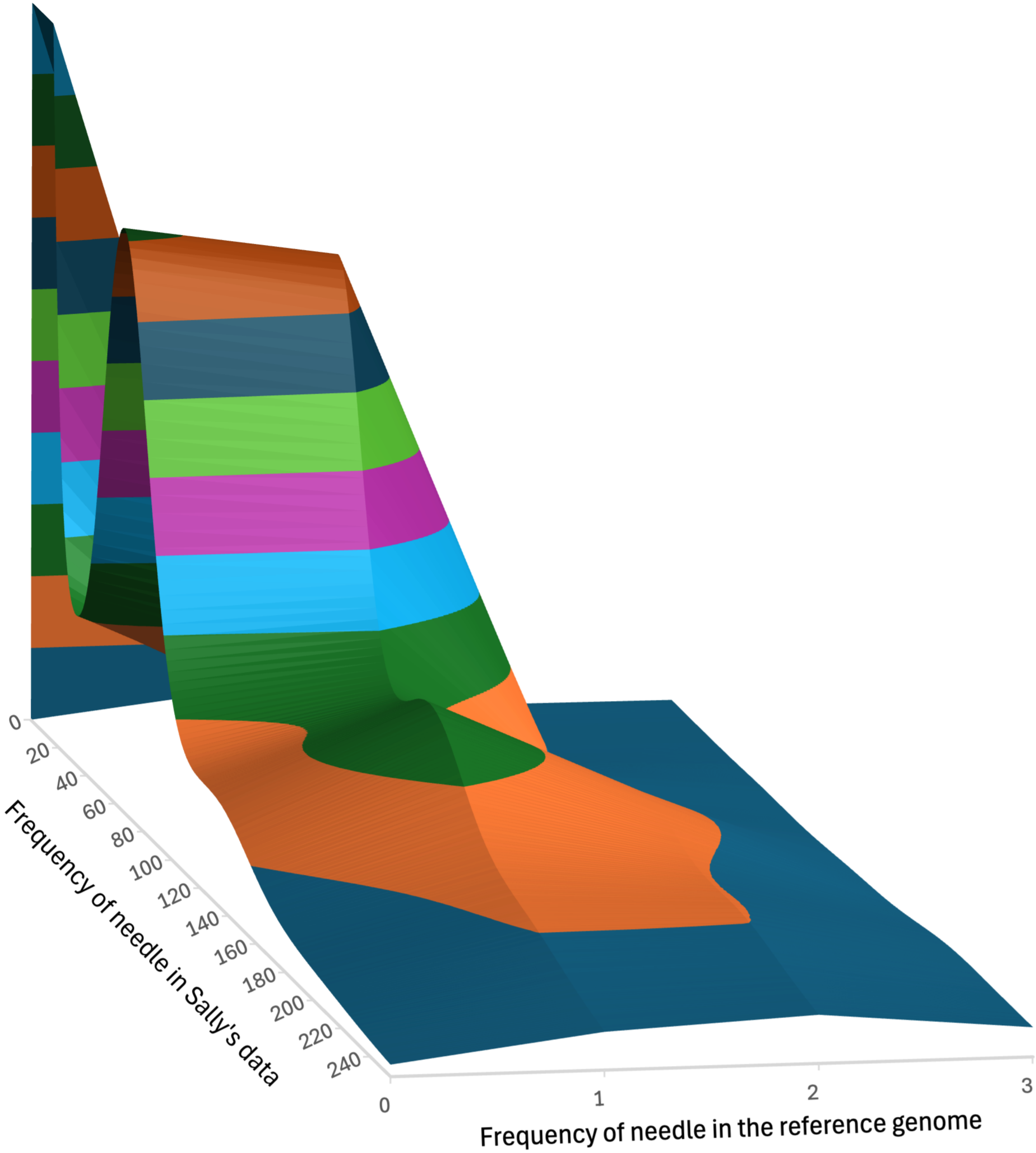

Figure 7: The joint frequency distribution between Sally's data and the reference genome.

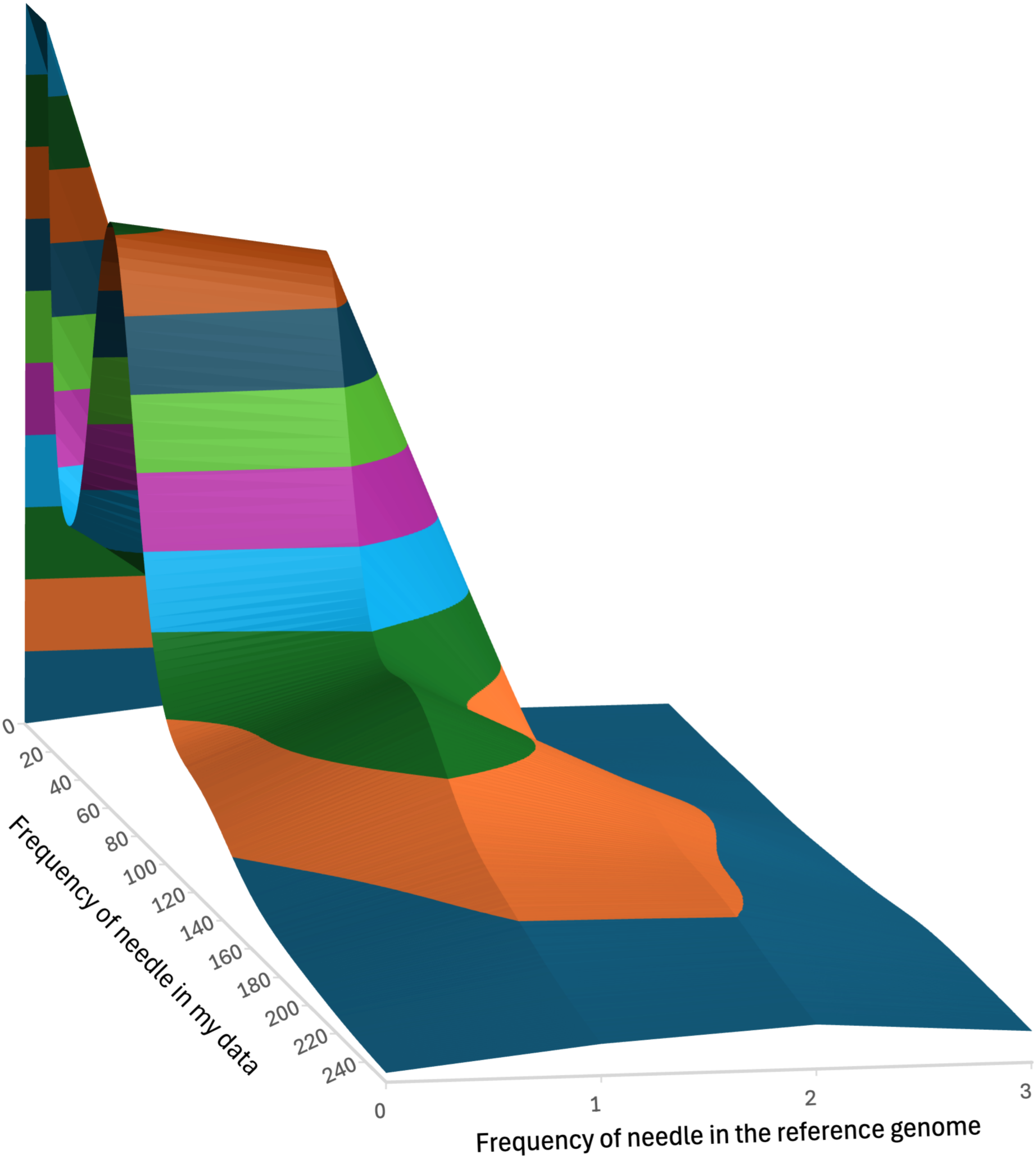

Figure 8: The same as Fig. 7 but for my data.

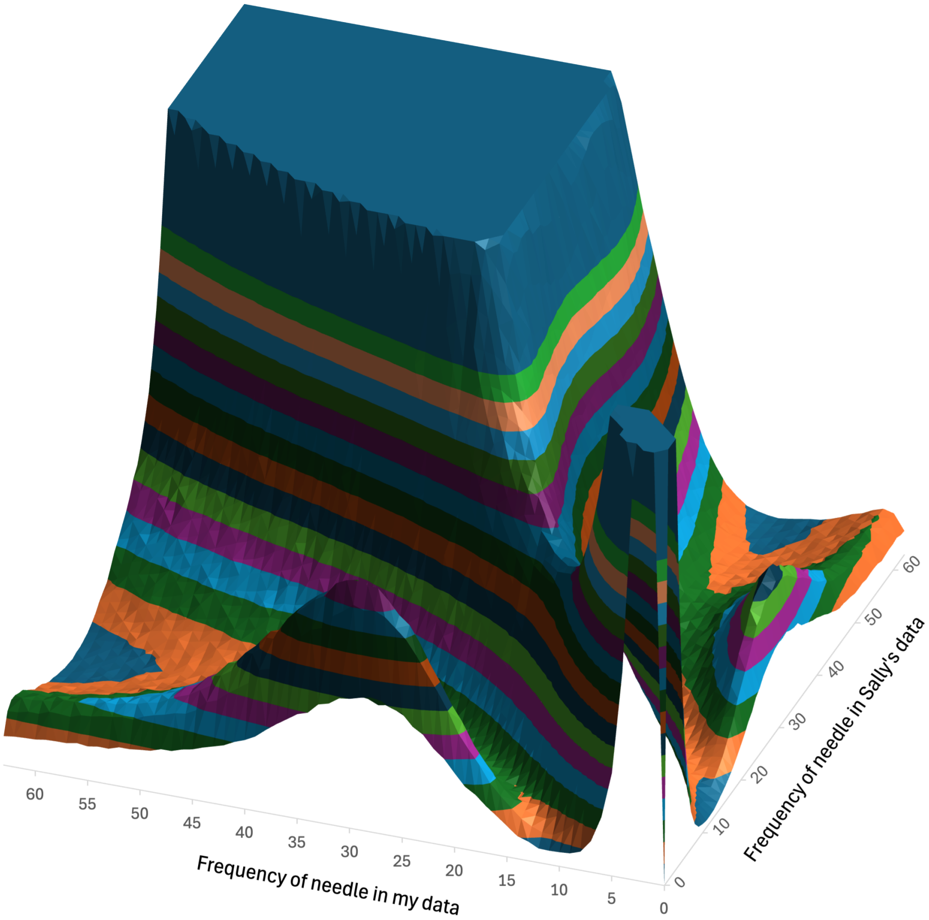

Figure 9: Zoomed view of Fig. 5 around the origin.

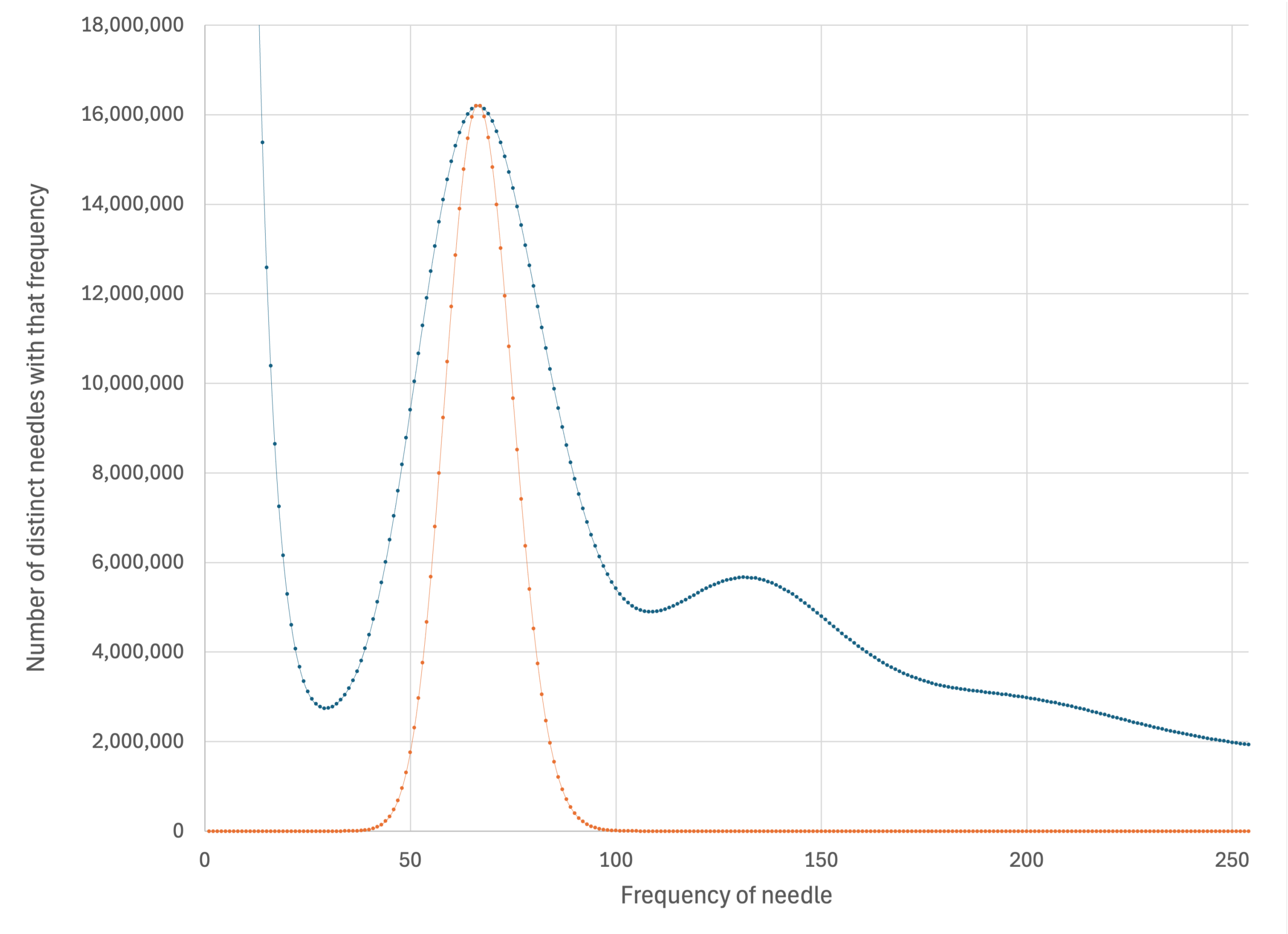

Figure 10: Fitting a Poisson distribution to the singles peak of Sally's data of Fig. 2.

10. The warts

As noted in Sec. 1, I confess here an omission from the above sections: once I realized that I didn't have any easy explanation for why I was getting far too many singles, I decided that I should go back to my throwaway line "described by a Poisson distribution" in Sec. 5, and actually fit Poisson distributions to the peaks in Fig. 2. I started with the main singles peak. The result was Fig. 10. I put that figure into this paper in a "Poisson" section, and just stared at it. Another disaster. I think it's worth including what I originally wrote into this paper, to show how far down the wrong rabbit hole I went:

> Standard physicist sanity checks confirm that there's nothing wrong with the Poisson distribution function that I used: the full width at half maximum of the orange curve is about 19.5, and we would expect it to be around $2.36\sqrt{66.5} \approx 19.2$. Rather, it's the experimental data that makes no sense: its full width at half maximum is about 41, or around twice as much as it should be for a mean count of 67.
>
> To put this into some sort of conceptual context, it is useful to think in terms of the *relative* width of the distribution—*i.e.*, the ratio of the width to the mean—which should be proportional to $1/\sqrt{n}$ for n independent samples. Being twice as wide for a given mean

> implies that we really only have the statistical power of *one-quarter* of the data that we think we have. In other words, we think that we're looking at around 67 independent random samples of the underlying sequence data, but in reality it's only got the statistical power of around 17 independent random samples.

So now I not only had a genome that seemed to be twice as long as it was supposed to be, halving the coverage of our raw data, but even *that* data seemed to have only a quarter of the statistical power it should have. I felt like I'd been hit by a 2×4.

I again assumed that I had made a mistake, and had somehow only processed a quarter of the data and then replicated it four times. Shit happens. But I quickly shot down that theory, as the distribution would look totally different. I then cast a stink eye in Nebula's direction, wondering if they had somehow only done a quarter of the sequencing and then fabricated the other three-quarters from that by random sampling. I mean, conspiracy theories aren't a usual part of science, but they are for me, and my day job *is* to help my teammates fight fraud and other forms of cybercrime. And it's not like we haven't had any recent precedent for suspicion [10]. The distribution would be right. But a short amount of thought convinced me that it wouldn't have been possible for Nebula to resample the data in the way needed to generate Fig. 10 without having four-times-longer raw sequences to start with, which made no sense if they were trying to do a dodgy and cut corners to fraudulently make money out of a service that they didn't actually perform in full.

Misquoting the guy living less than a mile down the road [11], I said to Sally, "I don't know what the fuck I'm doing."

Brain aching, I went back to trying to figure out what was going on with the doubled genome, as described in the above sections, only occasionally dipping back into this fourfold resampling problem. One bright light came from looking more carefully at Fig. 4: *it looked too thin.* Rotating it to look at it from overhead, in Fig. 11, its thinness was obvious. D'oh! (Proofreading this paper, I realize that the peaks of Fig. 9 are visible in Fig. 11 too. Double d'oh! I had Fig. 11 long before I had Fig. 9. Mumble mumble hindsight.) If the data was being wrongly quadruplicated—by me, or Nebula, or Russians, or whomever—it would spread out both Sally's data *and* my data—*independently.* The contour lines around the singles peak should be *circles*, not ellipses. That told me that the fourfold problem was *correlated between my needles and Sally's*, so it was not a data processing or resampling error. Indeed, those ellipses are half as wide as they are long, so if I somehow rotated the orange Poisson curve in Fig. 10 by 90° around a vertical axis through its peak *into the page*, and then another 90° around a horizontal axis to make its peak point in my direction, then it would fit across the peak of the ridge in Fig. 11 (vertically up the page, as we are looking at it) beautifully. That conceptual rotation doesn't really make any sense, but it reinforced for me that the fourfold distribution was *a property of the genome itself*, not an artifact of data processing.

Two steps forward, two steps back [12]: before long I was expending considerable energy creating a physically plausible model of a compound haploid genome that possessed within it four randomly-sampled copies of the 6.3-billion-base haploid genome (by now I had sorted *that* mess out).

It was pure fiction. The seed of my debunking it came when I was walking home from work through Franklin Park one evening, and realized that I could get a better look at that compound object if I just had more data. Buying more would take time and money. But I *already* had

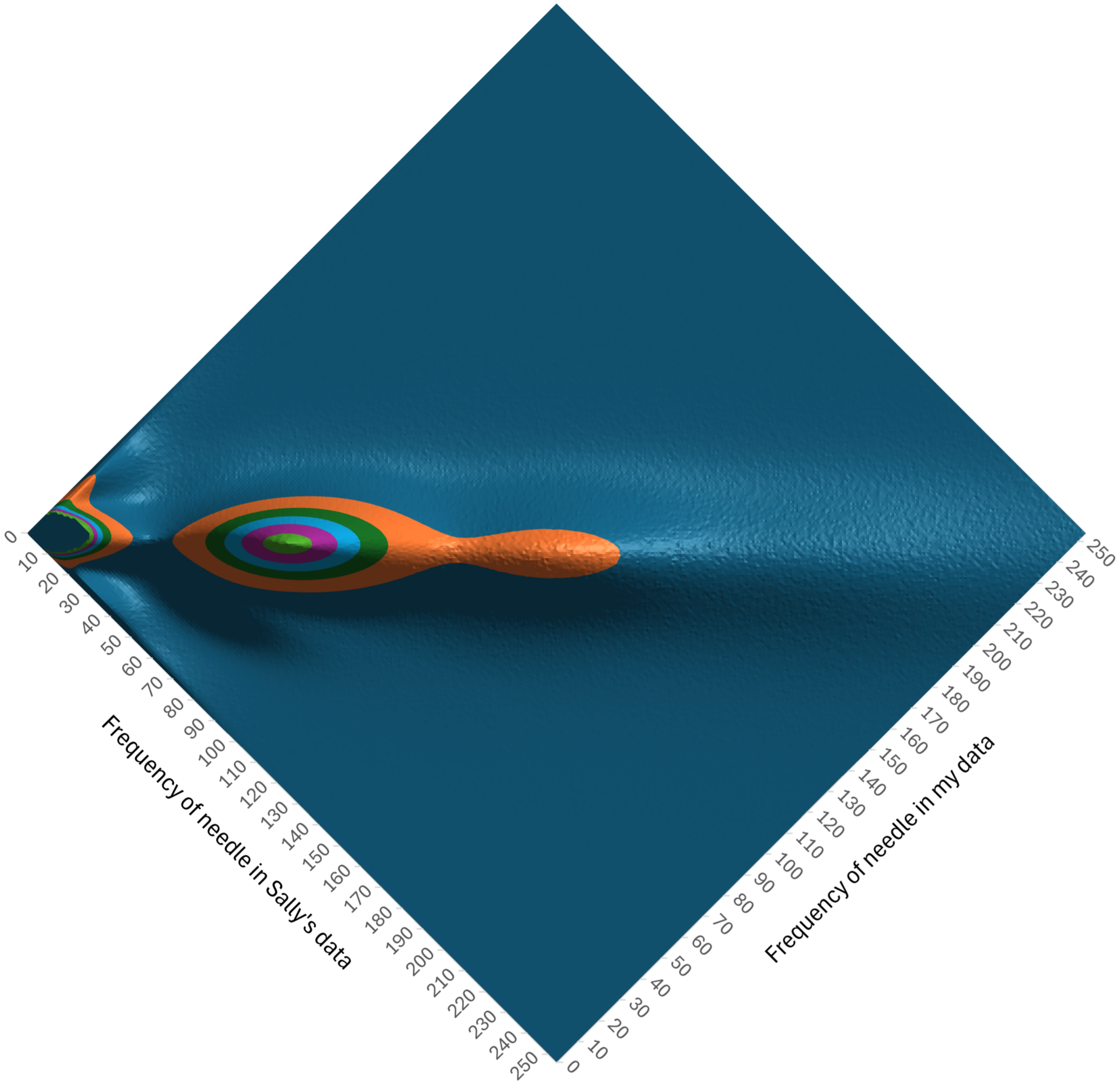

Figure 11: The same as Fig. 4, but looking straight down from overhead.

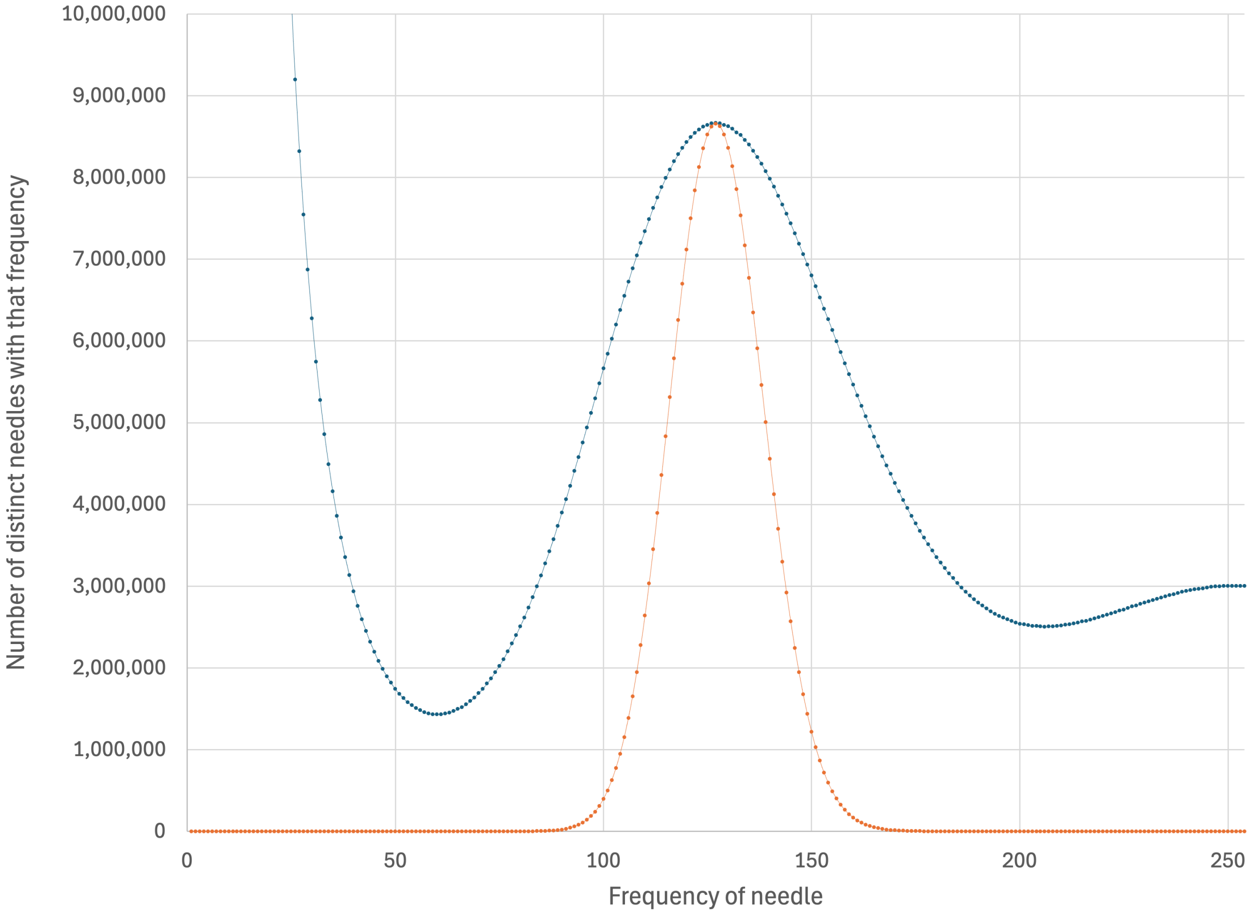

Figure 12: The equivalent of Fig. 10 for my data combined with Sally's.

more data: I could just *add my frequencies to Sally's* to create a "combined" dataset. Sure, it would have four haploid genomes in it rather than two, so more mutations to ultimately try to sort out, but at least it would give me a better look at the compound object: the sampling width should be reduced by a factor of around $1-1/\sqrt{2} \approx 29\%$—not a ton, but nothing to sneeze at.

It was trivial to create this combined dataset and compute the frequencies of its frequencies, which I show in Fig. 12. Excellent: the peak is around 127, which is very close to the sum of Sally's 66.5 and my 61.3. But *it is now 2.8 times as wide as the Poisson.* What the fuck?

It didn't take me long to realize that the Poisson *did* get 29% thinner, compared to its mean, *but our data didn't*, so the ratio went up from 2 to 2.8. So the problem *isn't* that we don't have enough sampling data: the object really *does* have that spread in frequency, and the amount of data we started with is more than enough to resolve it.

But . . . what could it mean, physically, for the genome to have a spread in frequency around the singles peak? That doesn't make any physical sense at all. If something appears exactly once in the haploid genome, then it can't appear 0.9 times, or 1.1 times. As the old saying goes, you can't be half-pregnant.

Then the penny dropped. The genome doesn't have that spread: *our measuring tool does.* It is well known (to everyone except me, but now me as well) that sequencing machinery does

Base	All positions	First position
A	29.85%	22.51%
C	20.36%	35.89%
G	20.46%	31.30%
T	29.33%	10.29%

Table 2: Unigram frequencies for my Nebula data, for all read positions and just the first position.

not sequence all parts of the genome equally well. There's a bunch of chemistry that gets in the way when the distribution of bases varies from the average in particular ways. The blue curve is spread out because some of the needles are sampled less often than average, and some are sampled more often than average, simply because of these "technical issues." It's not a uniform random sample across the genome, as I had assumed—implicitly—to that point: it's a *base-position-weighted* random sample.

So . . . a time-traveling interlude from the future here. I wrote the above words on August 29. On September 4 I told my best friend Greg Burnham that the last piece of the project—the sequence-growing algorithm that I will describe in the next section—*actually worked*, piecing together longer sequences from the individual needles. Around half of these short sequences matched parts of the reference genome exactly, *and the other half did not appear in the reference genome at all.* That was the final proof that half of our genome *really is* missing from the reference genome. (I mean, that's not a spoiler. I literally told you that above in the Abstract and in Sec. 1.)

Except . . . my estimate of how much the frequencies of the needles could vary from one needle to the next was off. I could see in the debug logs for the first part of the program that I wrote that the variations were about twice as large as what I got from my statistical model. I double-checked my calculation of the model, and it looked right. *That damn doubling of the width of the singles peak was still there*, even though I was sure that it would not be. Again, to quote something from the September 4 draft of this paper that I have since edited:

> It is probably fair to assume that the position-dependent bias (the technology-dependent thing that screwed me up in the last section) will be *the same* at that position in the DNA strand [the point of mutation] for the two different haploid genomes, since (if I understand correctly) it depends on the chemistry of bases in bulk, not the next few bases that happen to come along on the strand. If that's a fair assumption, then all we have left are the coin flips that determine which of the two haploid genomes happened to be chosen for each little 150-base sequence that Nebula's machine happened to sequence.

I don't actually know if it was fair to assume that. But I did. And now I knew that I was wrong.

I then had a horrible thought: could Nebula's sequencing machine *simply have a hard time cutting the DNA at certain bases?* Horrible, because such a bias is something that I could have measured—*should* have measured—weeks ago. Surely not.

It was trivial to test. I already had a program to compute the unigrams (frequencies) of bases from a needle frequency table—I wrote it just before I wrote the one to compute the bigrams for Table 1. With just a few changes I got it to only do that *for the first base of each 150-base Nebula read*, and ran it on my own data. I compare the two in Table 2.

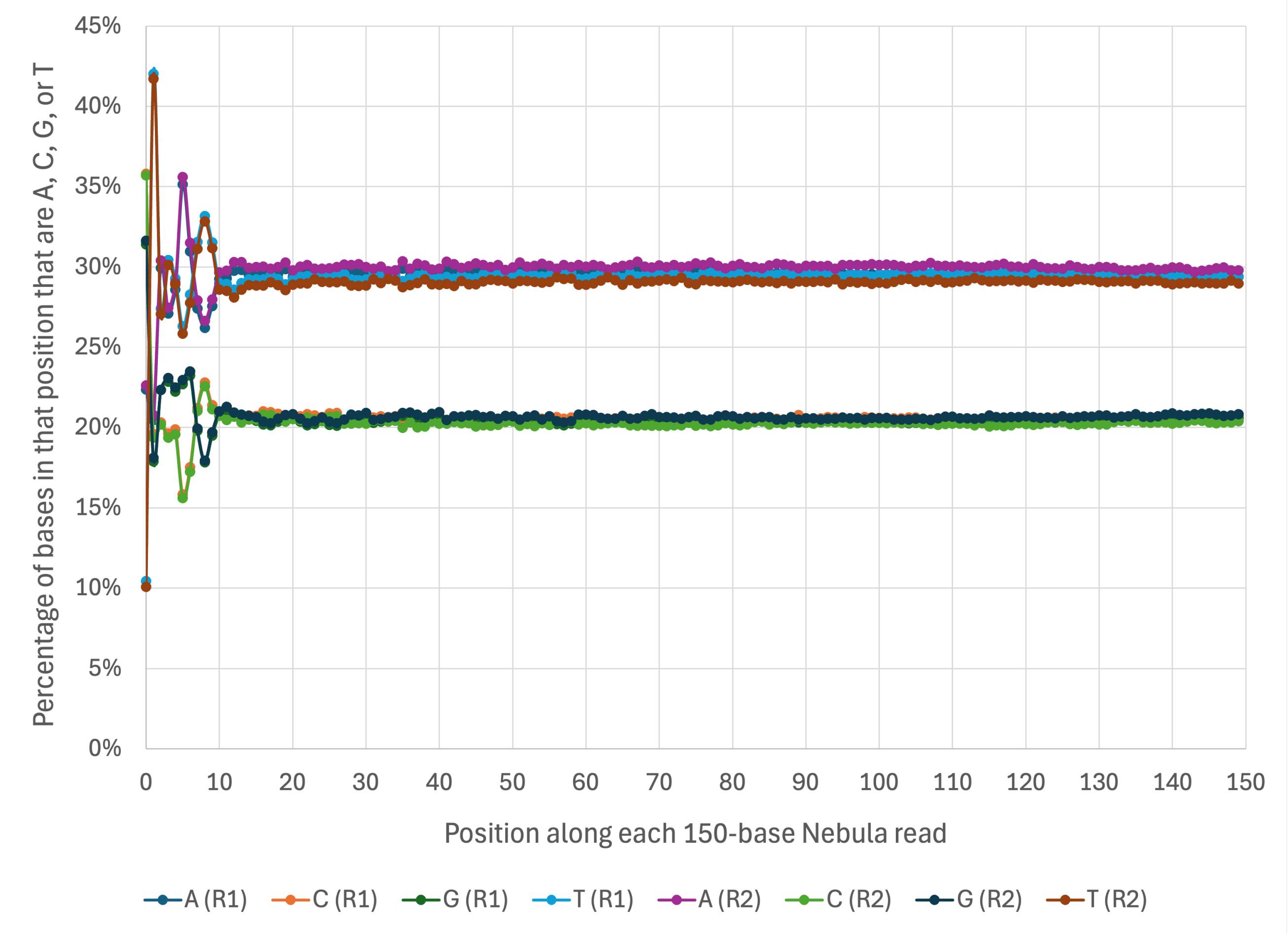

Figure 13: Percentage of each base across the 150 positions of each Nebula read, for Sally's data.

Motherfucker! It's completely off. Not just slightly off: `T` bases make up nearly 30% of the human genome [8], but they only start 10% of the Nebula reads! What a disaster. I'm an idiot.

But in this life, today's disaster is tomorrow's improvement. If I could figure out *why* the percentages in Table 2 are so off, I might have a chance of correcting for it.

The first thing I needed to do was check that I hadn't made a mistake in identifying my mistake. (You've gotta watch out for going into a recursive death spiral in this game.) I expanded the new program to compute the frequencies for *all* 150 positions of each Nebula read. I also kept separate the results for the R1 and R2 files, just in case there were any *more* funny buggers waiting to jump out at me from that direction. I show the results for Sally in Fig. 13.

I could only shake my head in disbelief. The first ten positions on each read are totally biased! I needn't have kept the R1 and R2 files separate: they follow each other closely, and indeed in Fig. 13 most of the R2 plots cover their R1 counterparts (due to the order that Excel plots them in), so that the R1 ones only occasionally peek out due to random noise. (I didn't make the dots and lines thinner, because already I can barely make out all the different colors; I needed help from non-colorblind Sally to even confirm that I was looking at the right ones.) That the R1 and R2 agree so closely also puts to bed any lingering doubts I had about reading the R2 file backwards: I definitely did not. I then did the same analysis for my own data, and got Fig. 14.

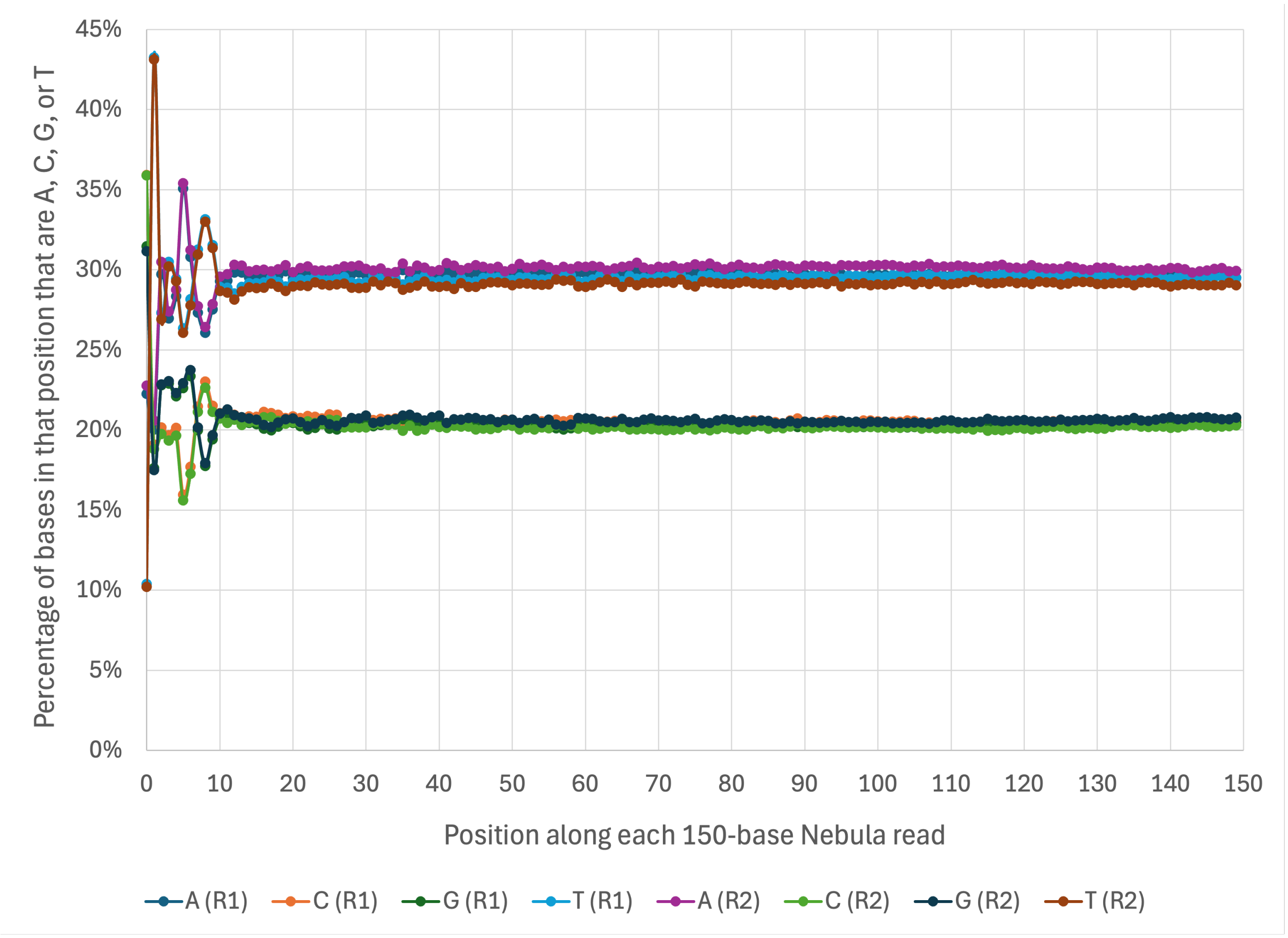

Figure 14: The same as Fig. 13 but for my data.

It's almost identical. This is clearly a "response curve" for the sequencing machines that Nebula uses. *Of course*, I realized, *it doesn't usually matter for anyone using this data.* Who the hell would care about precisely how many times any particular sequence appears in the data? All that matters, for any normal use of the data, is that the sequences appear many times (hopefully more than 100, if you were a skillful-enough cheek-swabber) so that you can reduce the effects of occasional sequencing mistakes. I'm using the data in ways for which it was not intended.

Well, fuck me. So how do I work around this newest blunder?

Just cut off those split ends. I mean, I do remember having hair, but even when I did, I don't remember ever having any concens about the split ends that Sally's conditioner bottle assures me it will fix, every time I am in the shower. Must be a girl thing. Reminds me more of the legendary Aussie (OK, maybe Kiwi) band of my early teenage years, who went on to even better things [13]. Anyway. Focus. Cutting these split ends off all of the 5,969,925,328 raw 150-base sequences in our Nebula source data files will take some hours, but such is life.

(Some hours later.)

The split ends have been trimmed off. Sally's cleaned master file has 3,125,048,271 sequences with a total of 436,951,308,122 bases; the latter is about 6.67% smaller than the total in Sec. 2 because we have discarded the first 10 out of every 150-base read. Mine has 2,852,926,662

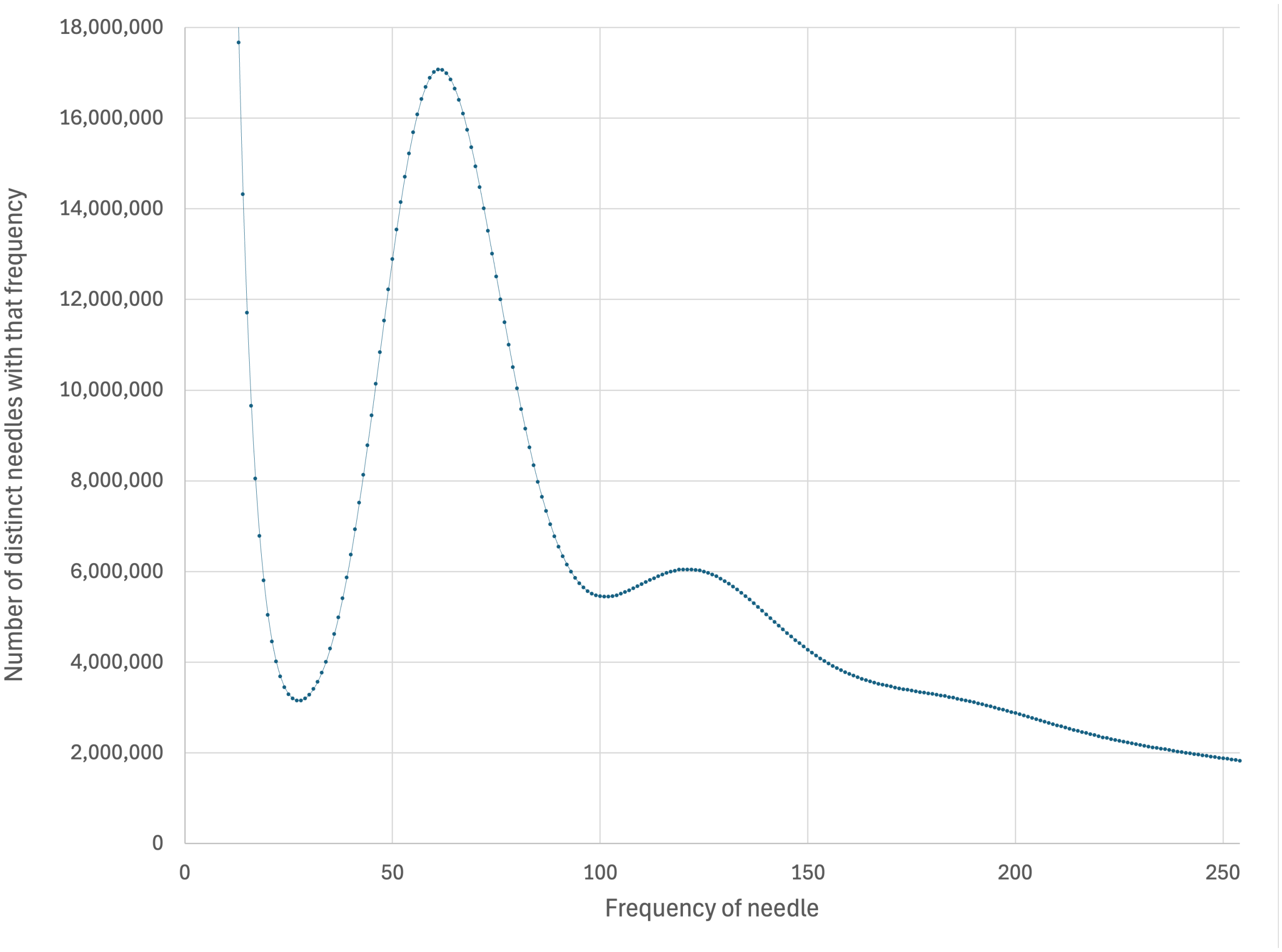

Figure 15: The equivalent of Sally's Fig. 2 but with only clean needles.

sequences with a total of 398,828,080,192 bases, with the latter again about 6.67% smaller than its counterpart in Sec. 2.

My first question was: would these "clean needles" (hmm, reminds me of the needle exchange program back in Melbourne, Australia, but anyway) have any significant effect on the analyses of the previous sections? I computed the frequencies *of* frequencies of Sally's cleaned data, like I did in Sec. 4 for the uncleaned dataset. The program reported that there are now 3,656,593,732 distinct needles amongst the total of 390,091,247,764 needles, and that the maximum frequency of any needle is 153,521,525. For me, there are now 3,635,275,273 distinct needles amongst the 356,049,894,000 total needles, and the maximum frequency for any needle is 143,753,524. I show the new version of Sally's Fig. 2 in Fig. 15.

Not much difference. The peak has moved down from 66.5 to 61.4, or a reduction of about 7.7%; this is slightly larger than the 6.7% reduction I would have expected. For me, the peak has moved down from 61.3 to 56.5, or 7.8%, again slightly larger than I expected. Having 61.4 singles every 390,091,247,764 needles implies that our estimate of the length of Sally's haploid genome has increased slightly, to 6.35 billion bases, and my 56.5 for every 356,049,894,000 needles implies that mine is 6.30 billion bases. However, the apparent precision of each of these estimates should be taken with a grain of salt, because I am computing it using the *mode* of the singles, not the mean, and without a precise mathematical model of the former I can't accurately estimate the latter. Much more robust, in contrast, will be the *difference* between our two estimates, which to leading order is not dependent on getting the mode–median issue sorted. The best way of looking at this is to consider the length of each of our *diploid* genomes. Our best estimate for Sally's is now 12.7 billion bases, and our best estimate for mine is now 12.6 billion bases. That would imply that Sally has 100 million more bases in her diploid genome than I do. That makes sense: her second X chromosome is about 100 million bases longer than my Y chromosome. So she is female and I am male. Good to know. The precision of that 100 million number is very rough; but, then again, so, too, apparently, is the length of the Y chromosome, which seems to vary by up to 30 million bases between men [14].

The 50-million-base difference in the length in our haploid genomes confused me when I first computed it, all the way back in Sec. 5, because at that point in time I thought it was the difference between our 6.3-billion-base *diploid* genomes. Now it is the difference between Sally's 6.35-billion-base *haploid* genome and *the average of my two haploid genomes*, one of which, from my mother Helena, has an X chromosome, which is 100 million bases longer than the other from my father John, which has a Y chromosome.

It is not just—again—awe-inspiring that analyzing these little sequences can tell us that the female diploid genome is 100 million bases longer than the male, but it is also further evidence in favor of the conclusion that the haploid genome *really is* 6.3 billion bases long. (I don't know if I'm ever going to fully convince myself that that is really true.)

Returning to the difficulty of modeling mathematically the frequency curves, it's clear that we're in no better position to fit a Poisson distribution to Fig. 15 than we were with my disastrous attempt of Fig. 10. The curve is still just as spread-out. But that makes sense: even if we trim off the split ends, hair is still hair. That the split ends were causing some grief as the needle slid across them is completely separate from the bulk properties of the hair itself.

But I'm getting ahead of myself: you don't even know that part of the story yet. This is still my time-traveling future self from halfway through the next section talking. So it's time to end this interlude, hand things back over to my other self, and get back to the main story.

11. A rudimentary algorithm for growing sequences

We've gone a long way just using our little 16-base needles, but it's time to piece them together into longer sequences. I know that the field of genomic data science has decades of experience creating incredibly powerful algorithms to do this, but since I'm ignorant of all that I'm just going to try something simpler to start with, to at least demonstrate that the data makes sense. Others can do it properly using all the proper software later.

As the posters at Facebook used to yell at me from random walls, "START SOMEWHERE." Sage advice, and it's remarkable how many people fail to. My first thought is that we should start with just *one* of our two individual datasets—and because Sally got more data from Nebula than I did, it's not just chivalry that makes me choose hers—and select a "seed" needle from it. From this seed we try to grow the sequence out, base by base, from both ends, like those fun crystal-growing activities we did in elementary school. To grow it to the right, we simply need to shift all the bases in the needle one place to the left, and then inspect the four possible needles that correspond to filling the newly-emptied rightmost position with one of the four bases. Conversely, we grow to the left by shifting the bases one place to the right, and filling the newly-emptied leftmost position. This is just a form of hexadecigram prediction, except that instead of just looking at the training data and choosing from the four possible needles the one that occurred most often, we're going to apply some more sophisticated business logic based on what we learned, somewhat painfully, in the previous sections.

But before we get to that, how do we choose the seed needle? There are many ways to skin this cat, but for my first attempt I think the simplest goal is to find a seed that falls somewhere along a sequence that appears just once in the unmutated haploid genome. Such a seed should provide the strongest possible backbone for growth to the left and right, since the signal should then be as clean as we could possibly get, because the sequence is either there or it isn't. Well—not quite: we're eventually going to hit either a mutation or the end of the DNA strand. The latter is a proper termination of the sequence. For the former, some of our raw sequence samples will go down the path corresponding to one of the two haploid genomes, and the rest will go down the other. On average it will be 50 : 50, but of course our raw sequencing data is just a sample. It is probably fair to assume (*déjà vu*) that the position-dependent bias will be *the same* at that position in the DNA strand for the two different haploid genomes, assuming it depends on the chemistry of bases in bulk, not the next few bases that happen to come along on the strand. If that's a fair assumption, then all we have left are the coin flips that determine which of the two haploid genomes happened to be chosen for each little 150-base sequence that Nebula's machine happened to sequence. If we denote the frequency of the last base on the unmutated haploid sequence before the mutation by n, then the number of samples belonging to a specific one of the two haploid genomes will follow a binomial distribution for n coin flips, so will have an expectation value of $n/2$ and a standard deviation of $\sqrt{n}/2$. For example, if we're tracking an unmutated haploid sequence in Sally's data that just before the mutation has a frequency of 64 (*i.e.*, 64 copies of it happened to be in the samples that Nebula sequenced), then we'd expect 32 ± 4 (I use the physicists' standard of one-sigma notation) of those samples to be of the haploid genome that she got from her mother Edith, and the rest to be from the one she got from her father Vic.

So how do we choose a needle that puts us on the best possible sequence that appears once in both the haploid genome that she got from Edith (henceforth, "the Edith") and the Vic? What we want to avoid is choosing a sequence that we *think* appears once in each of them, but then, to our (computational) dismay, we find—perhaps many bases down the garden path—that it's actually just a part of a much longer sequence that appears *twice* in each of the Edith and the Vic, but we just happened to land on a section of it that is a variant that only appears in, say, the Edith.

The only way that I can see to avoid this sort of outcome is to check what sort of a sequence any given seed would lead to, up to either the ends of the strand or a mutation. I think we would be best off casting as wide a net as possible, having the maximum possible number of candidates at the top of the recruitment funnel, and then devise as many selection criteria as required to successively filter down those candidates until we finally land on exactly one most favored candidate. (Can you tell that I work at LinkedIn?)

For the top of the funnel, my gut feeling is that we should start with all of the needles that fall between the two minima of Fig. 15. My program tells me that those occur at frequencies of 27 and 102, with the singles peak between them topping out at a frequency of 61. (I will use the vague term "my program" or "the program" for the rest of this paper, since I may refactor it into a different implementation after I publish this paper. I also exclusively use the data for clean needles from this point.) It finds that there are 747,293,925 distinct needles with frequencies that fall between 27 and 102 inclusive, comprising a total of 48,502,077,600 needles in Sally's data. The latter is about 12.4% of the total number of Sally's needles, which is greater than the 10% that I quoted in Sec. 6 because I have opened up the acceptance window all the way out to the minima around the singles peak, since I am confident that the algorithm below will be able to eliminate needles accidentally bleeding in from the "infrared divergence" on the left and the doubles peak on the right.

Now, even though I intend to use *just* Sally's data to create a *de novo* assembly, my program *does* also load data about the reference genome, so that I can spit out useful tidbits of information for myself while building it. I thought about using the reference genome in the business logic itself, but that would not only destroy any pretense at a *de novo* assembly, it could also limit the proper growth of the sequences, or send it down the wrong garden path, because we now know that the reference genome has serious problems. I mean, half the fucking thing is missing. So I load it into my program only for giving me some breaking news insights as I build it, as well as for some useful *a posteriori* comparisons *after* we've assembled Sally's genome. I'll get to those further below.

So, *purely for my own information*, the program tells me that 375,327,829 of the singles appear in the reference genome, or about 50.2% of them. That seems suspiciously close to 50%, so let's drill into it a bit more. The *total* number of singles in Sally's data that also appear at least once in the reference genome is 25,049,592,562, or about 51.6% of them. OK, that's a little more random. Going back to the former number, we can look at it as 375,327,829 of the singles appearing in the reference genome versus 371,966,096 that do not. The difference is 3,361,733. Now, is that small enough to imply that the true rate should be *exactly* 50 : 50, with this difference being just some sort of random sampling? We already did the math for that a page ago: a random 50 : 50 sampling of 747,293,925 singles would have a binomial standard deviation of about 13,668. So our observed split is actually 246 sigmas. This doesn't mean that this is *not* fundamentally a 50 : 50 random sampling, since we have all sorts of sources of noise in the above measurements. But it at least gives me more confidence that this is not just a data processing error.

However . . . it is difficult for someone who doesn't even believe the Zapruder film to be authentic to take *anyone's* word on anything—*even my own.* Flaunting my inner OCD, I also got my program to double-check that *the 50% of needles missing from the reference genome weren't just in there backwards.* I know, I know—we checked the bigram frequencies above, so that's not possible overall. But humor me for a moment: there's still a chance that *for these 12% of singular needles* there might be significant stretches of DNA that somehow got inverted. It does happen—according to the internet—but only at a rate estimated to be 0.6%.

Anyway, my program tells me that 210,991,572 of Sally's 747,293,925 distinct singles appear reversed in the reference genome, so already we know that that can't be the source of an answer. For comparison, 645,340,291 of her 747,293,925 distinct singles appear reversed in *her own* dataset, so about 86% of them, which is an intriguingly high percentage. But just to well and truly put this issue to bed, I got the program to further count up the number I *really* cared about: how many of her distinct singles *missing from the reference genome* are actually in there, but reversed. The answer is 104,409,604, or 49.5% of her overall 210,991,572 distinct singles that appear reversed in the reference genome. That makes sense: we consistently seem to lose half of everything if we just use the reference genome.

So, now that I've convinced myself, again, that I haven't completely screwed the pooch with my data processing, we can return to the task of architecting the sequence-growing algorithm. As I was saying before that interruption to our scheduled programming, my idea is that we grow the sequence out from the seed to the left and to the right one base at a time. For definiteness, I'll just describe building to the right; the left is just the mirror image. Again, let's denote the frequency of the seed by n. We shift the seed's bases one place to the left, fill the newly-emptied position on the right with each of the four bases `A`, `C`, `G`, and `T`, in turn, and look up the frequency of each new candidate needle in Sally's frequency table. If the gods are smiling on us, exactly one of the candidates will also have a frequency of exactly n. The decision is then a no-brainer: we choose that base to be the one to add on the right. If none of the four candidates have a frequency of n, then we have one of four possible physical situations: we reached the end of the strand; we hit a mutation; we have stumbled onto a needle that also appears elsewhere in the genome; or we have moved to a position on the genome that happens to have been sampled a slightly different number of times than the n samples we got for the seed.

Let's consider this last case first. We know that almost all of our raw data comes in little sequences that, after cleaning by trimming off the split ends, are 140 bases long (the only exceptions being those broken by naughty `N` base codes), from which we compute 125 needles. The seed overlaps n of these little sequences. Assuming that those little sequences are chopped from *random positions* from many source copies of Sally's genome (so the Oswald shredder analogy breaks down here; imagine instead that he cuts them up, freehand, with scissors) (of course, this *might* now be true, now that we have cleaned the needles—but wasn't when I first wrote this, which is what triggered that cleaning), then you can think of the raw sequences as being like n stalks of hay that I have randomly grabbed from the huge pile sitting on the fifth floor of the Texas School Book Depository (uh ... OK, shredded paper has now magically turned into hay, but bear with me) and tied together with twine into a sheaf. The needle sits in the middle of the sheaf—the part of it that is wrapped by my clenched hand—where all n of the haystalks happen to overlap for those 16 bases. Now, when I move my hand—the needle—one base to the right, there's a chance that my hand will move past the end of one or more of these n haystalks—any that I happened to grab the right ends of (the last 16 bases of). There's also a chance that I start overlapping one or more *new* haystalks, since (I didn't tell you this bit) the pile of haystalks has—while you were looking intently at my face for that last sentence—magically rearranged itself into many overlapping sheaves of haystalks forming a continuous but somewhat random "rope" of sheaves of haystalks, and as my hand slides to the right it clenches slightly varying numbers of these haystalks within the rope. (OK, I need to unclench and reclench to move one base to the right, but you get the idea.) We don't know in advance what the average number of haystalks along the rope will be, since that depends on the position-dependent bias in our sequencing machinery, and it likely will vary slowly as we move down the DNA strand, if there is some sort of change of the bulk chemistry that makes the machinery work better or worse. In any case, once we *do* get onto a rope—oh, and I also didn't tell you that the Book Depository is now full of all of this hay rope; it's such a tangled mess that it's not obvious how many separate ropes there actually are—we could *estimate* the average number of haystalks from all the needle frequencies that we have seen so far—or a small stretch of them, at any rate, if we find that the average value is varying is we move along the rope.

But let's leave aside estimating the average number of haystalks for the moment, and take Oswald down to the second-floor lunch room and buy him a Coke from the vending machine as grateful thanks for his assistance. (Dammit, I can't report my success to that nice attorney for the Pepsi-Cola Company—who seems to have such grand plans for young Lee—because apparently he's already flown out of town. I'm going to blame my delay on all those road closures when I call him later in New York.) The real question is: by how much could the number of haystalks change each time we move down the rope by one base? Well, the sheaf is n haystalks wide before we move the needle, and practically every haystalk is 125 needles long. If, instead of a random mess of them within the rope, they were actually *evenly* spread out, we would fall off the end of a haystalk every $125/n$ base positions, on average, and we would likewise hit the start of a haystalk every $125/n$ base positions as well. For example, imagine that n is currently 63, so that the current sheaf is 63 haystalks wide. Then if we had (artificially, by hand) arranged the haystalks by placing each of them on the floor exactly two bases to the right of the last one we put down, then by the time we go to put down the 64th one, we will have moved 126 bases from where we started, which is past the end of the first one that we put down, which means that this arrangement gives us an almost-constant width of 63 bases (with the exception only being that one position at the end before the 64th haystalk—sorry, even *I* can't make 125 an even number). So, for this artificial evenly-laid-out scenario, our needle will fall off the end of a haystalk every two bases, and hit the start of a new haystalk every two bases (except for that odd one every 63 haystalks).

So, if on average the needle hits the end or start of a haystalk—henceforth, a "transition"—every $125/n$ base positions, we can equivalently think of it as hitting each type of transition at a *rate* of $n/125$ transitions per base; for the example above, a rate of about half a transition per base. Every "up" transition increments n, and every "down" transition decrements it. If we now drop that artificial evenly-spread-out scenario, and instead spread the haystalks out *randomly*, then the up transitions and the down transitions will each become Poisson processes, effectively independent because we will have practically forgotten where each particular haystalk started by the time we get to the other end. (Well, I will have. Sally can testify that I don't even remember what I ate for dinner last night.) One Poisson process adds to n, and the other subtracts from it, and so it is the *difference* of these two Poisson processes—a Skellam distribution—that describes the change of n from base to base.

So how large could Δn be for a one-base move to the right, in absolute value, for us to be comfortable that the result is still consistent with the model of a rope of sheaves of haystalks?

To answer that, remember that we want to follow the "diverse slate approach" when choosing our seed needle by dumping as many candidates into the top of the funnel as possible, even if we think that most of them don't have a hope in hell of getting the job, secure in the knowledge that they will be weeded out by our selection criteria further down the funnel—because we might occasionally find that our prejudice was actually wrong. So we are looking to find the value of $|\Delta n|$ that is still possible with some small but finite probability, such that we don't prematurely knock out a candidate sequence simply because it had a rare but possible coincidence of haystalk ends at one particular position. We can afford to be generous in our leeway, confident that our later selection criteria will still select out the best candidate.

I don't have a precise criterion for finding this threshold, but my gut feeling is that we should set the probability at 1 out of the 747,293,925 singles. We therefore just need to find the value of $|\Delta n|$ for which the two tails of the Skellam distribution have this combined probability. The answer is $|\Delta n| \approx 10$ at the highest transition rate we will encounter of 102 needles per 125 bases, $|\Delta n| \approx 9$ at the modal transition rate of 61 needles per 125 bases, and $|\Delta n| \approx 7$ at the lowest rate of 27 needles per 125 bases. Considering the dodginess of my probability threshold, I'm willing to just set the maximum $|\Delta n|$ to 10, to be safe, with the promise that I will model it in more detail one day. Maybe.

So, as we grow out the sequence from each candidate seed, we will continue growing to the right as long as one of the four candidate needles has a frequency that is within 10 of the last one. (When I first tested this hypothesis, with my program, I found that there were sequences that were clearly actually correct—they appear in the reference genome!—for which $|\Delta n|$ would be up to 20 in size, rather than 10. Fail fast! That is what caused me to time-travel back to the last section. So, good news: there is only one Michael J. Fox version of me from this point forwards.) We should also impose the restriction that each needle must be within the domain of frequencies (here 27 to 102) that we are taking to be possible singles. When either of these conditions fails to be true, we assume that we have hit one of the three other physical conditions described above (end of strand, mutation, or a needle that happens to appear somewhere else), and we terminate growing the candidate seed's sequence to the right, because all that we want *at this stage* is the longest stretch of "clean" backbone sequence that we can find.

So how are we going to grow these candidate seed sequences in practice? Well, I think that we should grow the two ends (right and left) simultaneously, which in practical terms means that we alternate sides for each base we attempt to add, at least until one of the sides terminates. But now consider a potential problem: what if we end up landing on a needle *that we already used in the sequence?* We would end up doing a Groundhog Day by repeating the same cycle over and over. (Have you figured out my favorite movie genre yet?) Not good. But that doesn't really make sense, because we started by trying to find singles, *which by definition only appear once in the haploid genome.* Once we use such a single, *we should remove it from our frequency table.* Doing that will prevent any endless cycles.

However, each of these sequences is only a *candidate* for selecting our first seed. Only one of them is going to win. So we need to store not just the bases that we add to each end of the sequence, but also their frequencies, so that we can put them back into our frequency table after we select the winner. Storing the frequencies has the nice by-product that we could then potentially do some statistical analysis of that set of frequencies for each candidate seed, and use that as a tiebreaker in our selection criteria for seed sequence candidates of equal length; for example, the smallest mean value of $|\Delta n|$ might be the tiebreaker.

Now, just *when* should we put all these frequencies back into the frequency table? It might seem that we should do it after finishing growing each candidate seed's sequence, because we want to give all candidates an equal opportunity. A laudable goal in general, but in this particular case it's not actually the right answer, because all of the singles that we accept for use in the first candidate seed's sequence *should* be removed from the pool, *because the sequences that we would grow for each of them is exactly the same sequence that we just built.* It doesn't matter *which* single we consider the "seed": any of them is sufficient to specify the sequence that they all fall on. Indeed, we don't want to build them over and over, because that would have a time cost that is quadratic in the length of the sequence, for absolutely no benefit.

Thus when we build the sequences for the candidate seeds, we should greedily deplete the frequency table for singles as we go. By doing so we will knock out many other candidate seeds, but they will have just joined the "teams" of those that happened to be chosen first. It really doesn't matter who the "captain" is.

Once we no longer have any candidate seeds remaining, we use the selection criteria to choose a winner. This is where some skill in crafting business rules will probably come in. Above I suggested that the length of the sequence should be the primary criterion, with mean $|\Delta n|$ being the secondary. I will likely refine and finesse these business rules after publishing this paper, so check out my web page [4] for whatever my latest thoughts are. Whatever they are, we select a winning candidate seed needle.

We then put all the frequencies for all the candidate seed sequences back into the frequency table, and throw away those sequences. All that we keep is our chosen seed. From it we now grow out the best *actual* sequence, including catering for mutations that split the Edith and the Vic, and catering as best we can for needles that also appear elsewhere in the genome. I'll describe both of these shortly. Once we find that we cannot grow it any more, we have our first actual sequence. We put it to one side, cherishing and guarding it. We do *not* put its frequencies back into the frequency table. Those needles are now permanently gone, into the locked box of our final chosen sequences.

We then rinse and repeat the entire candidate-building and winner-growing process, until there are no singles left.

If we do this right, then we might get close to "peeling off" all the sequences that appear just once in the haploid genome. That would be good enough for a first crack at the data, and I would be more than happy with it. But we might actually get a bit luckier. We've only targeted seeds that are singles—the low-hanging fruit—but after we remove them from our dataset, we *might* find that some of the needles that were originally doubles are now singles—*if* we manage to craft an algorithm that can deal with doubles as well as singles—and so we can keep going, peeling them off too, and so on. It's like a tech company who wants to get rid of the worst 12% of their employees. All they need to do is "stack rank" them all, and lay off the bottom 12%. Mission accomplished! But it's kind of like the bean hopper of a coffee grinder: once you grind the bottom 12% of beans, all the other ones shuffle down under gravity. You then grind the same amount out for your next shot. After eight shots, you're pretty well out of beans.

Maybe not the smartest move for a tech company. Never listen to the bean-counters. But it would be a brilliant result for *us*: imagine if we managed to extract *all* of the sequences like this? The only thing that would stop us is if the bean hopper is so badly designed that its funnel's cone is not steep enough, so that the force of gravity isn't enough to overcome the friction between the beans, and the bottom layer of beans often collude to obstruct the funnel like a bunch of formation skydivers. (Not wanting to impeach anyone, but I'm looking at you, DeLonghi. For the 2013 Magnifica XS model, anyway. The 2021 La Specialista is much better. But more than five years after I tried to return the damaged one on January 6, 2021, thwarted by the UPS Store being closed due to the posted First Amendment Event nearby, we're really getting sick of that squealing noise that its replacement makes on every shot [15].) I suspect that we will get to that sort of pileup with the highly repetitive sequences in the genome; after all, the maximum frequencies I reported back in Sec. 10 for the clean needles imply that at least one particular needle appears around 2.5 million times on each of our haploid genomes. That will require a bit more finagling to figure out, but those parts aren't as immediately interesting as the more unique sequences anyway, so I'm not worried if we can't untangle them in the first instance. We have to start somewhere.

Let's now return to the question of how the *actual* sequence-building algorithm should differ from that of the candidate seed sequences. Our main goal is to not simply give up when we can't find a needle whose frequency isn't within 10 of the last one, but rather to deal with some of the edge cases. The two most important ones are mutations between the Edith and the Vic, and needles that happen to also appear elsewhere on the genome. Let's first concentrate on mutations. We laid down the basis for most of this above: we expect $n/2 \pm \sqrt{n}/2$ of them to go down the Edith, and the remaining ones to go down the Vic. So all we need to do is inspect the frequencies of the four possible needles *in pairs*, and see if any of those pairs have frequencies that sum to $n \pm 10$ (our criterion from the Skellam distribution) and have a difference that is consistent with a standard deviation of $\sqrt{n}$ (from this binomial split between the Edith and the Vic). So, again, what do we deem to be "consistent with"?

In this case it's not clear to me if false positives or false negatives are more harmful to the fidelity of our grown sequences, but it turns out that we can come to a relatively simple working rule with a bit of analysis. My initial inclination is to allow the mutation split decision unless we are pretty sure that it is wrong, because we *do* have one trump card left: we expect most mutations to be either single-nucleotide variants or indels, in which case our traversal of the Edith or the Vic should be relatively short before we again see needles at the full (unsplit) frequency of around n. So we are safe to, for example, apply a confidence interval that only gives us a 5% chance of making *any wrong decision at all for the whole genome*, as long as we then "probe" each such provisionally-allowed mutation split, to check that we do indeed "get to the other end" of the split between the Edith and the Vic. If we don't, then we can give up, back out of that mutation split decision, and move to the next stage of the decision waterfall (which will be to check if the needle is actually a double). But if we *do* see each of the Edith and the Vic returning to full unsplit frequency, then they should again start predicting *the same needle* on the other side of the mutation. If that isn't true, then, again, we back out of mutation split decision, and go back down the waterfall.

That this algorithm using hexadecigram needles allows us to automatically "resync" the Edith and the Vic on the other side of the mutation with essentially no computational effort—unlike, say, the standard Unix `diff` tool, which has to perform extra analysis to perform a resync—is a nice bonus. No Sherlock Holmesing is required. Elementary, my dear Warren.

With all of these "safeties" in place, then, we can construct a confidence interval for acceptance of the mutated frequencies that almost certainly will not reject any legitimate mutations. The internet tells me that about 0.4% of the genome is understood to vary between the haploid genomes. Assuming that the same is true for the "shadow genome"—the half missing from the reference genome—then that rate implies that we can expect to see about 25 million mutations in the full 6.3-billion-base haploid genome. Using the Bonferroni correction, we can be 95% confident of not making any mistakes at all if the probability of error on each test is 5% divided by 25 million, or about 2×10^{-9}; we assign 10^{-9} to each tail. The cutoff frequencies are now easy to compute from the inverse cumulative binomial distribution. For $n = 61$, say, the resulting confidence interval runs from 7 to 53. This is so wide that we would actually get a tighter bound simply by requiring that twice the frequency of the Edith and twice the frequency of the Vic each fall within the domain of 27 to 102 that we set for our pool of candidate singles in the first place. On top of that we can impose the requirement that twice of each of them falls within 10 of the frequency n that we last had. Essentially, we can scrap our statistical analysis of the mutated frequencies completely, and just apply our original policies to their corresponding doubled counterparts. There may be a more correct way of performing this analysis, but this will be good enough for me for a first approximation.

All that is left now is to figure out how to deal with doubles, or multiples in general. We get down to this part of the waterfall if none of the four candidate needles has an acceptable full-strength frequency near n, and nor does any pair of those candidate needles. We now want to check if any of them could feasibly be a double: namely, that we are currently looking at a part of the rope where it appears, but it has exactly one other appearance somewhere else on the genome. This is a more difficult situation to untangle, because we don't know where that other position is, so in general the position-dependent machine bias could be completely different. All that we can check is whether any of the four candidate needles has a frequency such that, when n is subtracted from it, the result is again a valid single, namely, between 27 and 102. We should also include our ±10 for haystalk transitions. Putting this all together, our criteria are then that the candidate needle frequency must be at least $n+17$ and no greater than $n+112$, where we we can narrow the range slightly if n itself falls within 10 bases of the edge of its own allowed domain of 27 to 102. For example, if $n = 50$, and none of the four candidate needles has a frequency between 40 and 60, and no pair of candidate frequencies fulfills the criteria described above for mutation subsequences, then we check if any of the four needles has a frequency that is between 67 and 162. This is a range that is getting dangerously close to n itself, and it may not be feasible to peel off a single sequence from a string of doubles except in the luckiest of circumstances.

The last situation we need to address is when we find that more than one of the four candidate needles (or more than one pair of them, for the mutation criteria) fulfills our statistical criteria. I think that we have no option but to try all alternative solutions, because if the statistics tells us that they're all possible, then we don't really have grounds for choosing one over another. We need to guard against an exponential explosion of possibilities, of course, but a capped amount of exploration is probably the best course.

I suspect that this is the most that I will be able to assemble of Sally's genome using this rudimentary algorithm. So now to build it, and see.

12. Results

When I finished writing that last section, yesterday, September 7, my intention was that I would simply include an overview here of whatever results I managed to get before publishing this paper, since I know that there is probably a lifetime of fun that I could have using my program to understand and construct sequences from our data. I had already—on September 4 (I originally wrote "a week or so earlier," until I checked with the daily copies of this paper that I send to Greg Burnham, which shows you that my sense of time is currently shot)—conceived of a way of showing, in a single huge heatmap diagram, how the sequences I would construct could be compared to the reference genome, although I wasn't sure if I'd be able to assemble long-enough sequences to make the details in the diagram large enough to be seen. I'm confident that I can write the code to build that diagram, because I know someone with a good set of libraries for resizing arbitrarily-large floating-point images into ones small enough that I could include one in this paper. But the journey of building those sequences with the program last night has been so fascinating that I don't believe that I will get to writing that program this week. So, on this Star Trek birthday (they beat me into the world by 16 days), since I have already been showing you "how I've been making the sausage" in such detail that you're probably nauseous by now, I may as well include my original description of my envisaged heatmap diagram, and then scrap my plans for actually including it here in the next three days, so that I can just tell you about the fascinating journey itself:

> We can also perform a visual comparison of these sequences to those in the reference genome by conceptually creating a scatterplot of the "position" of every needle in Sally's assembled sequences to the "position" of the corresponding needle in the reference genome, *if it exists in there and occurs only once*; otherwise, we plot no point for that needle in Sally's assembly. Of course, these "positions" are not absolute, depending as they do on the arbitary concatenation of the set of sequences into a single long string of bases, delineated in the data file by header lines, which represent "joins" between physically distinct objects; *i.e.*, the joins have no actual basis in physical reality. But *within each sequence* the order of the bases is meaningful. If any sequence found here corresponds for any substantial number of bases with one in the reference genome, then we would expect to see a short diagonal line with unit gradient in the graph. These short diagonal lines would also be robust against variants of relatively short length, since even indels would create an imperceptible change from a perfect correlation. Of course, these diagonal lines would not represent *every* correspondence between Sally's assembly and the reference genome, but any amount of verification would be valuable. The result is shown in Fig. X.

The concept here of the "position" of a needle in both the output file of Sally's partially-assembled sequences, and in the reference genome data file from the NIH, is actually so useful for generating those tidbits of "cheat" information for myself while building the sequencing program that I will expand on it some more here. I went back and modified the program that I described in Sec. 9 that generated the frequencies for the reference genome, including ambiguous base codes, to also collect these "positions," into another one of my 16 GiB lookup tables. Each of the 2^{32} possible 16-base needles has one of three possible statuses in the reference genome: it could be absent; it could appear exactly once; or it could appear more than once. If it is absent, I store a particular sentinel value (actually `0xffffffff`) in the lookup table. If it appears more than once, I store a different sentinel value (actually `0xfffffffe`). If it appears exactly once, I store its "position," as simply the number of bases from the start of the reference genome data, as described in the above quote. There can actually be more than one needle that is listed as being at the same position (I'm going to drop the scare quotes on "position" from this point, although it's still not a real position) if there are ambiguous base codes, as the program computes all of the possible variants and gives each of them the same position. This 16 GiB lookup table gives me a way to efficiently look up whether any given needle within Sally's partially-assembled sequences appears in the reference genome, and, for those cases in which it appears exactly *once* (which I'd hoped to be a fair proportion of the time, because we *are* concentrating on likely singles), the actual position in the reference genome.

So now for those "cheat" tidbits. I have built a "debug print" functionality into my codebase that lets me print out things while I'm debugging the code, which I then remove before shipping it to my website. But in this particular case the output from it has already influenced my thinking about the genome so much that I'm going to share it here. None of these example are terribly significant, but I include a few to show some properties that might be giving some hints (maybe) of what is actually going on with the reference genome.

I show the first example in Fig. 16, and describe the format there in its caption. This one is about as vanilla as it comes. Even though I *only* use Sally's data to construct it, you can see from the reference genome positions that it matches a sequence on there exactly, between (what I'm labeling as) positions 773,477,904 through 773,477,924. Even the base that the program couldn't

```
====================
A: 116        -
C:   0        -
G:  48 1,913,075,767
T: 114   773,477,903
--------------------
A:  64   773,477,904
G:  61   773,477,905
T*  61   773,477,906
T*  61   773,477,907
T*  61   773,477,908
C*  61   773,477,909
G*  61   773,477,910
A*  61   773,477,911
G*  61   773,477,912
T*  61   773,477,913
T*  61   773,477,914
T*  61   773,477,915
C*  61   773,477,916
A*  61   773,477,917
A*  61   773,477,918
T*  61   773,477,919
G*  61   773,477,920
C*  61   773,477,921
T:  62   773,477,922
C:  61   773,477,923
C:  62   773,477,924
--------------------
A:  33        -
C:   7        -
G:   0        -
T: 107   773,477,925
====================
```

Figure 16: An example candidate seed sequence. The middle section represents the seed sequence that the program successfully grew from the seed needle. The top section represents the base immediately to the left of this successful sequence that the program was unable to add. Likewise, the bottom section represents the unsuccessful base immediately to the right of the successful sequence. The first column is just the base in question. The second column is "`*`" for the 16 bases of the seed needle and "`:`" otherwise. The next column of numbers is the frequency of the needle that each base came from; obviously, all 16 bases from the seed needle have the same frequency. The last column of large numbers is the position of that needle in the reference genome, with "`-`" if the needle is missing, or "`(multiple)`" (not shown in this example) if it appears more than once; as noted, this column is *not* used in the sequence-growing process, but is simply included for my edification. Returning to the top and bottom sections, the four rows show the four possible bases that were considered at each end, with details of the corresponding needles created by appending each base to the other 15 bases from the sequence already grown.

decide on at each end is actually there: we see that positions 773,477,903 and 773,477,925 would be correctly filled if the program had just chosen `T` for each of them. Of course, the "cheat" data is for my eyes only: the program has a virtual Chinese Firewall that prevents it from leveraging the reference genome position data in its selection algorithm. The program's rules of engagement didn't allow it to choose these `T` bases because they jumped up to frequencies of 114 and 107 respectively, and the candidate seed sequence logic (at least, as I currently have it programmed, per the arguments in the previous section) does *not* try to peel singles from doubles—which the two actually-correct (cheating from the reference genome) surrounding needles clearly now are, appearing (I infer) elsewhere on Sally's genome with frequencies 50 ± 10 and 45 ± 10 respectively. Of course, on the left side, there's also the `A` candidate with a frequency of 116, which is an equally good choice as `T` with 114. *We* know that that would break the run of reference genome positions, but the algorithm does not, again because of that Chinese Firewall.

In Fig. 17 I show an equally vanilla example of the opposite kind. Here *none* of the needles appear in reference genome. The algorithm managed to grow `CA` on the left and `T` on the right, and then got stumped because each of the next base positions had two equally-good possible candidates: `C` (68) and `T` (61) on the left, and `A` (50) and `T` (61) on the right. Again, my candidate seed sequence algorithm currently doesn't allow it to explore both possibilities, per the arguments of the previous section. In any case, it is heartening that *what started on the shadow genome stays on the shadow genome*. If I had seen sequences flipping and flopping between the reference genome and the shadow genome, I'd know that something was wrong with my analysis; that outcome would not make any sense physically. Never the twain shall meet, right?

Well . . . not really. If they *never* met, then where would the shadow genome be hiding? It's not like the cell nucleus has a storage unit at Extra Space Storage that it can store its shadow genome in, apparently not needing it for its day-to-day business. I'd be worried if I inferred that the shadow genome was floating around the cell nucleus untethered, apparently invisible to our sequence alignment algorithms for some unfathomable reason.

Fortunately, *I do* occasionally see the twain meeting. I show one example in Fig. 18. It seems to be nicely traveling along from at least position 1,937,064,233 through to position 1,937,064,238 on the reference genome (only getting stuck at the left end because there are two other equally-good candidates there), then seems to jump to positions 275,068,645 and 275,068,646 on the reference genome, and then goes off the reference genome completely. I have no idea what is going on with those two positions in the transition, but it feels to me like the reference genome has somehow glued two parts of the genome together that shouldn't be. I show a similar but cleaner example in Fig. 19 where there is no extra excursion in the transition. I *will* say that, in my *extensive* tens of minutes of experience looking at these sequences, most such "twains" are clean breaks. Conversely, I *do* see other curious jumps in sequences that are all on the reference genome, such as the one I show in Fig. 20, where now there seem to be *multiple* jumps between different places on the reference genome.

As I told Greg last night, I get the strange feeling that *I might actually have the teacher's edition of the book*, with all the answers *and* fully-worked solutions printed in it, rather than the other way around. That might be wishful thinking or just pure arrogance on my part, but I'm struggling to reconcile any other explanation with the data I'm seeing.

In any case, as I said, I think I could spend the rest of my life trying to figure out all these sequences—even if I crafted better algorithms. So I'll draw a line under it for now. You can always check out whatever results I subsequently report on my website [4]. I hope to create

```
=====================
A:   5        -
C:  68        -
G:   1        -
T:  61        -
---------------------
C:  65        -
A:  64        -
A*  61        -
A*  61        -
A*  61        -
C*  61        -
C*  61        -
T*  61        -
A*  61        -
G*  61        -
C*  61        -
G*  61        -
T*  61        -
T*  61        -
C*  61        -
A*  61        -
A*  61        -
T*  61        -
T:  60        -
---------------------
A:  50        -
C:   2        -
G:   1        -
T:  61        -
=====================
```

Figure 17: A candidate seed sequence that does not appear on the reference genome at all.

```
=====================
A:  56  2,932,075,453
C:  54  1,937,064,233
G:   2        -
T:  54    391,754,484
---------------------
T:  54  1,937,064,234
T:  55  1,937,064,235
G:  55  1,937,064,236
A:  58  1,937,064,237
G:  63  1,937,064,238
C:  65    275,068,645
C:  68    275,068,646
T*  61        -
T*  61        -
A*  61        -
C*  61        -
A*  61        -
T*  61        -
A*  61        -
C*  61        -
C*  61        -
C*  61        -
C*  61        -
A*  61        -
T*  61        -
C*  61        -
C*  61        -
G*  61        -
C:  62        -
A:  66        -
T:  61        -
C:  59        -
T:  60        -
A:  63        -
G:  59        -
G:  59        -
C:  60        -
C:  61        -
T:  58        -
G:  58        -
C:  57        -
A:  59        -
G:  67        -
C:  67        -
---------------------
A:  12        -
C: 171    (multiple)
G:   5        -
T:  89        -
=====================
```

Figure 18: A candidate seed sequence that appears to be on the reference genome and then goes off-reference, after a curious two-base excursion.

```
====================
A: 168    (multiple)
C:   2       -
G:  83       -
T: 322    (multiple)
--------------------
G*  61       -
C*  61       -
A*  61       -
G*  61       -
A*  61       -
T*  61       -
T*  61       -
T*  61       -
C*  61       -
A*  61       -
A*  61       -
T*  61       -
C*  61       -
T*  61       -
G*  61       -
C*  61       -
G:  52       -
A:  43 1,861,035,320
G:  44 1,861,035,321
A:  47 1,861,035,322
T:  46 1,861,035,323
--------------------
A:   2       -
C:   0       -
G: 147 1,861,035,324
T:   0       -
====================
```

Figure 19: A cleaner example of a sequence that starts off-reference and then goes on-reference.

```
=====================
A: 150    (multiple)
C:  35        -
G:  63  1,747,425,154
T:   4        -
---------------------
A:  92  1,353,941,776
A:  85  1,353,941,777
A:  79  1,353,941,778
G:  79  1,353,941,779
T:  80  1,353,941,780
G:  77  1,353,941,781
T:  69  1,105,608,652
T:  71  2,879,009,967
C*  61  2,879,009,968
G*  61  2,879,009,969
G*  61  2,879,009,970
A*  61  2,879,009,971
G*  61  2,879,009,972
C*  61  2,879,009,973
T*  61  2,879,009,974
G*  61  2,879,009,975
G*  61  2,879,009,976
T*  61  2,879,009,977
G*  61  2,879,009,978
G*  61  2,879,009,979
A*  61  2,879,009,980
T*  61  2,879,009,981
G*  61  2,879,009,982
A*  61  2,879,009,983
T:  64  2,657,205,807
---------------------
A:  61    536,283,448
C: 183        -
G:  87  1,218,668,196
T:  54  2,657,205,808
=====================
```

Figure 20: A sequence that seems to jump between different places on the reference genome.

actual partially-assembled sequence data for Sally and for me, and maybe for both Sally and me in that rolled-together dataset that I created in Sec. 10.

But I will make one more comment (OK, yes, I admit that I pre-rolled the first version of this one a couple of days ago): it is straightforward for you to confirm, *explicitly and without any computational reliance or trust in me whatsoever*, whether any sequence that I find Sally or I to have, which I claim the reference genome does or does not have, *really is* either present or missing from the reference genome. For example, take the sequence GGACTCTTATAGTTACCA, which is just one randomly-selected seed sequence I saw early on in Sally's data, which my program told me also appears in the reference genome. It's easy for any of you to just go onto the command line and do

```
$ grep -in GGACTCTTATAGTTACCA GCA_000001405.29_GRCh38.p14_genomic.fna
```

to prove for yourself that it's there on line 23,199,678. Of course, there was a finite chance that this sequence happened to be broken by a newline in the NIH's FASTA file (it turns out that this one isn't), so you can either do some command-line foo to work around that, or else you can make use of the "dump" file that I created directly from the reference FASTA file for unambiguous base codes that only has a newline when the sequence breaks (the code for which is so short that you can easily confirm that I'm not playing any funny buggers with it), or else you can give up altogether, take my word for it, and play the statistical odds by rolling the dice with a different sequence instead.

As a converse example, take the 25-base sequence AATCCCAGTGGCGTCATACTGCATA. My program tells me that its seed needle appears in Sally's data 61 times, so once every 6.3 billion bases. (I start with candidate seed needles right on the singles peak, so it's no accident that I have these ones with a frequency of 61 lying around in my early play with the data.) Of course, this longer 25-base sequence will appear fewer times, because it corresponds to a different family of *n*-grams, namely, the 25-grams, which I didn't compute (and until Apple offers me a laptop with at least four petabytes of memory, I won't be doing any time soon either). Leaving that to one side, for the moment, I can check for this 25-base sequence in *my own* raw data files from Nebula, explicitly, via

```
$ gzcat HYMQHR3VV_R1.fq.gz | grep -in AATCCCAGTGGCGTCATACTGCATA
```

In this case it takes hours for it to tell me that it appears 20 times in just this first raw R1 file. The equivalent for my R2 file,

```
$ gzcat HYMQHR3VV_R2.fq.gz | grep -in AATCCCAGTGGCGTCATACTGCATA
```

tells me that it appears 24 times there as well. So overall it appears 44 times in my raw data. Or again, you can use my dump file, directly from the FASTQ files,

```
$ gzcat john.fq-dump.txt.gz | grep -in AATCCCAGTGGCGTCATACTGCATA
```

which also gives 44 hits, which hopefully gives you some confidence that my dump files are faithful to the raw data and not corrupted by Russian interference. Now, going back to Sally, I can do the same with *her* raw Nebula R1 and R2 files,

```
$ gzcat H48ZYY71_R1.fq.gz | grep -in AATCCCAGTGGCGTCATACTGCATA
$ gzcat H48ZYY71_R2.fq.gz | grep -in AATCCCAGTGGCGTCATACTGCATA
```

to find that this 25-base sequence happens to appear 59 times in her data. So 61 times for her 16-gram drops down to 59 times for her 25-gram, and my 25-gram appears in my data 44 times. That's more than consistent with it appearing on each of our haploid genomes exactly once; *i.e.*, it's probably not a mutation, or if it is, it's one that we both happen to have on both of our haploid genomes.

You can now look for this 25-base sequence explicitly in the reference genome:

```
$ grep -in AATCCCAGTGGCGTCATACTGCATA GCA_000001405.29_GRCh38.p14_genomic.fna
```

Or, again, you can use my dump file, which doesn't have the newlines:

```
$ gzcat ref.fna-dump.txt.gz | grep -in AATCCCAGTGGCGTCATACTGCATA
```

It's not there. Of course, there are occasional ambiguous base codes in the reference genome that map to multiple possible bases, and this `grep` method would fail to match them. But the frequency table I created from the reference genome *included all of the possible alternatives* for the ambiguous base codes, so my program is not going to miss them—only this simple manual `grep` double-checking might, occasionally.

And this is just *one* example—from untold millions of cases that I will ultimately be able to give you—that both Sally and I both have on *both* of our haploid genomes—which means that Edith had it, and Vic had it, and Helena had it, and John (my father; I was John Paul, for disambiguation, until he died) had it—and so it is more than likely shared by just about everyone of European ancestry, given how many different European countries those four parents or their forebears came from. *But the reference genome doesn't have it.*

13. Conclusions

It took an astounding amount of incredible work to compile the bible of our genetic code—possibly the most impactful project ever undertaken in any field of science. For someone with the initials J. C. to come along and claim that *there's a second book* is not a decision to be taken lightly. I've seen how that story ends.

But all that I've actually shown is that *Sally and I* have a second book. It's possible that aliens modified our DNA and gave us that second book, so that everything in this paper is only true for the two of us. But I think that that theory is highly unlikely. We're not related. We both did 23andme in 2014, and again in 2019 when they came out with the v5 chip, both so that we could get better reports but also so that we could upload the results into MyHeritage DNA. In February 2023 I uploaded mine into Genomelink, and in January 2024 into Sequencing.com and GEDmatch, and then did Ancestry DNA from scratch because they wouldn't accept the 23andme data. I also redid MyHeritage DNA from scratch at the same time. And, of course, in September 2024 we both did the whole genome sequencing from Nebula. None of these companies have reported anything anomalous about our genomes. We have relatives on both sides of our families that are reported as matches at the expected degree of genetic overlap. Ancestry DNA matched my daughter Jayde at the expected rate of 50%.

Lest it be argued that all of these except Nebula are superficial, in 2011 my son Jack had genetic testing to confirm the diagnosis that he is on the spectrum. The medical staff were so excited to detect the variant in his DNA that confirmed this diagnosis that they asked his mother to ask me for a genetic sample as well. I can't recall if they never sent the kit or else never told me the results, but the point is that they went deep into his genome, for medical purposes rather than direct-to-consumer entertainment. Geek gene aside, they didn't report anything anomalous with Jack's genome.

I conclude from all of this that Sally's genome and my genome are fundamentally no different than anyone else's. We're not a different species. Which means that *everyone has a second book of genetic code, that for some reason has been overlooked to date.*

Believe it or not [16].

References

[1] J. P. Costella, `johncostella.com/aliens` (2025).
[2] NIH, Genome assembly GRCh38.p14 (2022).
[3] C. M. Gooding Jr. and T. C. Mapother IV, *Jerry Maguire* (1996).
[4] J. P. Costella, `johncostella.com/genome` (2025).
[5] R. A. Millikan, *Science* **32** (1910) 436.
[6] R. A. Millikan, *Phys. Rev.* **2** (1913) 109.
[7] A. Piovesan *et al.*, *BMC Res. Notes* **12** (2019) 106.
[8] S. Scherer, Guide to the Human Genome (2010).
[9] M. Caton, *The Castle* (1997).
[10] U.S. Department of Justice, *United States v. Elizabeth A. Holmes, et al.* (2022).
[11] D. J. Trump, `x.com` (2025).
[12] P. J. Abdul, *Opposites Attract* (1989).
[13] Split Enz, *I Got You* (1980).
[14] A. Rhie *et al.*, *Nature* **621** (2023) 344.
[15] J. P. Costella, `amazon.com` (2021).
[16] J. Scarbury, *Believe It Or Not* (1981).