Source : https://berthub.eu/articles/posts/part-2-reverse-engineering-source-code-of-the-biontech-pfizer-vaccine/

Reverse Engineering Source Code of the Biontech Pfizer Vaccine: Part 2

All BNT162b2 vaccine data on this page is sourced from this World Health Organization document.

This is a living page, shared already so people can get going! But check back frequently for updates.

Translation: Français / 日本語

In short: the vaccine mRNA has been optimized by the manufacturer by changing bits of RNA from (say) `UUU` to `UUC`, and people would like to understand the logic behind these changes. This challenge is quite close to what cryptologists and reverse engineering people encounter regularly. On this page, you'll find all the details you need to get cracking to reverse engineer just HOW the vaccine has been optimized.

I thought this would just be a fun puzzle, but I have just been informed that figuring out the optimization procedure & documenting it is tremendously important for researchers around the world, as this would help them design code for proteins and vaccines.

So, if you want to help vaccine research, do read on!

The leader board

Here are the current best entrants to the optimization algorithm (average of 20 runs):

Name	Codon Match	Nucleotide Match	Author	Comment

codon mapping	79.51 %	91.52 %	Harry Harpel	A simple sta codon mapp
most-frequent.py	78.57 %	91.08 %	Seo Sanghyeon	Codon frequ optimization python_cod
dnachisel	76.99 %	91.06 %	Erik Brauer	DNAChisel a
dnachisel	76.89 %	90.89 %	Pedro José Pereira Vieito	DNAChisel a
remap	71.11 %	88.59 %	Howard Chu	Map every c an amino ac the best cod that amino a
3rd-cg.py	60.83 %	85.11 %	Peter Kuhar	If third posit already 'G' c change. Oth replace thir position by a protein still r done. Other a G.
3rd-gc.go	53.06 %	81.55 %	bert hubert	If third posit already 'G' c change. Oth replace thir position by a

				protein still r done. Otherw a C.
dnachisel	46.33 %	79.48 %	Naomi Jacobs	DNAChisel a
NOP	27.63 %	72.23 %		Does not do optimization

BioNTech

We should all be very grateful that BioNTech has shared this data with us. And of course we should also be grateful to the many many researchers and lab workers that worked for decades to bring the state of the art to the point that such a vaccine could be developed. It is marvelous.

Because it is so marvelous, I want to understand everything about the vaccine. I wrote a page Reverse Engineering the source code of the BioNTech/Pfizer SARS-CoV-2 Vaccine that describes in some detail what is in the mRNA of the vaccine. It helps to read this page before continuing, I promise you it will be interesting.

The post left open some questions however, and this is where it gets fascinating.

The codon optimization

The vaccine contains RNA code for a very *slightly* modified copy of the SARS-CoV-2 S protein.

The RNA code of the vaccine itself however is *highly* modified from the viral original! This has been done by the manufacturer, based on their understanding of nature.

And from what we understand, these modifications make the vaccine **much much more** effective. It would be a lot of fun to understand these modifications. It might for example explain why the Moderna vaccine needs 100 micrograms and the BioNTech vaccine only 30 micrograms.

Here is the beginning of the S protein in both the virus and the BNT162b2 vaccine RNA code. Exclamation marks denote differences.

```
Virus:   AUG UUU GUU UUU CUU GUU UUA UUG CCA CUA GUC UCU AGU CAG UGU GUU
Vaccine: AUG UUC GUG UUC CUG GUG CUG CUG CCU CUG GUG UCC AGC CAG UGU GUG
               !   !   !   !   ! ! ! !     !   !   !   !   !
```

RNA is a string (literally) of RNA characters, `A`, `C`, `G` and `U`. There is no physical framing on there, but it makes sense to analyse it in groups of three.

Each group (called a codon) maps to an amino acid (denoted by a capital letter). A string of amino acids is a protein. Here is what that looks like:

```
Virus:   AUG UUU GUU UUU CUU GUU UUA UUG CCA CUA GUC UCU AGU CAG UGU GUU
          M   F   V   F   L   V   L   L   P   L   V   S   S   Q   C   V
Vaccine: AUG UUC GUG UUC CUG GUG CUG CUG CCU CUG GUG UCC AGC CAG UGU GUG
               !   !   !   !   ! ! ! !     !   !   !   !   !
```

Here we can see that while the codons are different, the amino acid version is the same. There are 4*4*4 codons but only 20 amino acids. This means you can typically change every codon into one of two others, and still code for the same amino acid.

So in the second codon, `UUU` was changed to `UUC`. This is a net addition of one ‘C’ to the vaccine. The third codon changed from `GUU` to `GUG`, which is a net addition of one `G`.

It is known that a higher fraction of `G` and `C` characters improves the efficiency of an mRNA vaccine.

Now, if that was all there was to it, this could be the end of this page. “The algorithm is change codons so we get more G and C in there”. But then we meet the 9th codon which changes `CCA` to `CCU`.

Throughout the ~4000 characters of the vaccine, this happens many times.

Our challenge

The goal is: find an algorithm that modifies the ‘wild type’ RNA code into the BNT162b2 one. Because everyone would like to understand how to turn viral RNA into an effective vaccine. The algorithm does not need to reproduce the *exact* RNA code of course, but it would be super nice if it came up with something very similar, while also being brief.

To help you, I have provided the data in a number of forms, as described on the GitHub page.

Note that in these files the `U` mentioned above appears as a `T`. `U` and `T` are the RNA and DNA manifestations of the same information.

The easiest place to start might be the 'side-by-side.csv' file. This lists the original and modified version of each codon, side by side:

```
abspos,codonOrig,codonVaccine
0,ATG,ATG
3,TTT,TTC
6,GTT,GTG
...
3813,TAC,TAC
3816,ACA,ACA
3819,TAA,TGA
```

There is also an equivalency table that shows wich codons can be interchanged without changing the amino acid output. Please find this in codon-table-grouped.csv. There is also a visual version here.

A sample algorithm

On the GitHub repository you can find 3rd-gc.go (and 3rd-gc.py).

These implement a simple strategy that works like this:

If a virus codon already ended on G or C, copy it to the vaccine mRNA

If not, replace last nucleotide in codon by a G, see if the amino acid still matches, if so, copy to the vaccine mRNA

Try the same with a C

Otherwise copy as is

Or in `golang`:

```
// base case, don't do anything
```

```
our = vir

// don't do anything if codon ends on G or C already
if(vir[2] == 'G' || vir[2] =='C') {
        fmt.Printf("Codon ended on G or C already, not doing anything.")
} else {
        prop = vir[:2]+"G"
        fmt.Printf("Attempting G substitution, new candidate '%s'. ", prop)
        if(c2s[vir] == c2s[prop]) {
                fmt.Printf("Amino acid still the same, done!")
                our = prop
        } else {
                fmt.Printf("Oops, amino acid changed. Trying C, new candidate '%s'. ", prop)
                prop = vir[:2]+"C"
                if(c2s[vir] == c2s[prop]) {
                        fmt.Printf("Amino acid still the same, done!")
                        our=prop
                }

        }

}
```

This achieves a rather poor 53.1% match with the BioNTech RNA vaccine, but it is a start.

When you design your algorithm, be sure to only base your choices on the virus RNA. Do not peek into the BioNTech RNA!

If you have achieved a score beyond 53.1% please email a link to your code to bert@hubertnet.nl (or @PowerDNS_Bert and I'll put it on the leader board at the top of this page!

Things that will help

As with every form of reverse engineering or cryptanalysis, it helps to understand what we are looking at.

GC ratio

We know that one goal of the ‘codon optimization’ is to get more `C`s and `G`s into the vaccine version of the RNA. However, there is also a limit to that. In DNA, which is also used to manufacture the vaccine, `G` and `C` bind together strongly, to the point that if you put too many of these ‘nucleotides’ in there, the DNA will no longer be replicated efficiently.

So some modifications may actually happen to manage *down* the GC percentage of a stretch of DNA if it was getting too high.

I tweeted about this earlier.

Codon optimization

Some codons are rare in human DNA, or in certain cells. It may be that some codons are replaced by other ones simply because they are more frequently used by some cells.

I tweeted about this earlier.

RNA folding

We’ve been looking at codons up to here. The RNA itself however does not know about codons, there are no markers that say where a codon begins and ends. The first codon on a protein however is always ATG (or AUG in RNA).

RNA curls up into a shape. This shape might help evade the immune system or it might improve translation into amino acids. This only depends on the sequence of RNA nucleotides and not on specific codons.

You can submit RNA sequences to this server of the Institute for Theoretical Chemistry at the University of Vienna and it will fold RNA for you. This is a very advanced server that does meticulous calculations.

This Wikipedia page describes how this works.

It may be that some optimizations improve folding.

I am also told that this paper by Moderna (another mRNA vaccine manufacturer) may be relevant: mRNA structure regulates protein expression through changes in functional half-life.