**Updated Code to locate Single Nucleotide Polymorphisms (SNPs) and Overlapping G-Quadruplex Sequences (G4s) for Multiple Genes**

**Abstract**

G-quadruplexes are non-canonical DNA secondary structures that form when guanine-rich sequences fold into a four-stranded structure stabilized by Hoogsteen base pairing. Recent studies have shown that some G-quadruplex-forming sequences can hinder DNA structure stability and can control gene regulation by impairing DNA damage repair and transcription. Single nucleotide polymorphisms (SNPs) are genetic variations that arise within DNA sequences, contributing to the genetic diversity observed between individuals. While most SNPs are non-harmful, some can be linked to various diseases. To investigate the potential role of SNPs in causing G-quadruplexes, we developed a new Python code that automatically computes the distance between SNPs and the closest G-quadruplex complexes for multiple genes. Additionally, the code integrates the G4 Hunter tool to detect G-quadruplex sequences in genes with high scores. Overall, the updated code provides a more comprehensive approach to identifying the overlaps between SNPs and G-quadruplex sequences which could lead to a better understanding of the possible significant role of SNP in the formation/maintenance of G-quadruplex structures in disease development.

1. **Introduction**

Small genetic variations in the form of single nucleotide polymorphisms (SNPs) are highly prone to occur within nucleic acids of the double helix DNA, which can significantly impact gene regulation and protein function. While SNPs are common and crucial for genetic diversity, some can lead to clinical manifestations [6, 9]. In addition, the planar arrangement of four guanine bases linked by non-Watson-Crick hydrogen bonds, known as G-quadruplexes, can stabilize the secondary structure of DNA and also contribute to clinical manifestations. The discovery of the guanine tetrad-forming sequence motif was a result of early studies on the self-assembly of guanylic acid, where double helix B-DNA form can fold into G-quadruplexes [5,8]. There is a noteworthy correlation between SNPs found in G-quadruplex sequences and the expression levels of the corresponding gene [1]. Furthermore, it was discovered that somatic mutations situated in the 5’ untranslated regions (UTR) modify the stability of RNA G4 and consequently have an impact on gene expression in cancer patients [10]. As the topic of G4-quadruplexes gains significant attention, numerous methods have been developed to computationally predict the locations of G4 quadruplexes, with G4Hunter being one of the most recent ones. This algorithm uses a sliding window approach and takes into account the G-richness (the fraction of Gs in a sequence) and G-skewness (the G/C asymmetry between the complementary strands) to identify regions with a mean score above a certain threshold. Empirical evidence demonstrated the high accuracy and sensitivity of this approach compared to other methods that deal with only G-richness and a consensus sequence for G4-quadruplex [2]. As Recent literature began to rely extensively on the G4Hunter tool in an attempt to explore the functionality of the G4-quadruplexes [3,4]. Nevertheless, there has not been any published work that integrates this tool to identify overlaps of SNPs and G4-quadruplexes and contribute to the topic of the association between them. Thus, this work aims to provide a Python script that automatically maps the distances of SNPs for various genes with the distances of their respective g4-quadruplexes predicted by the G4Hunter tool.

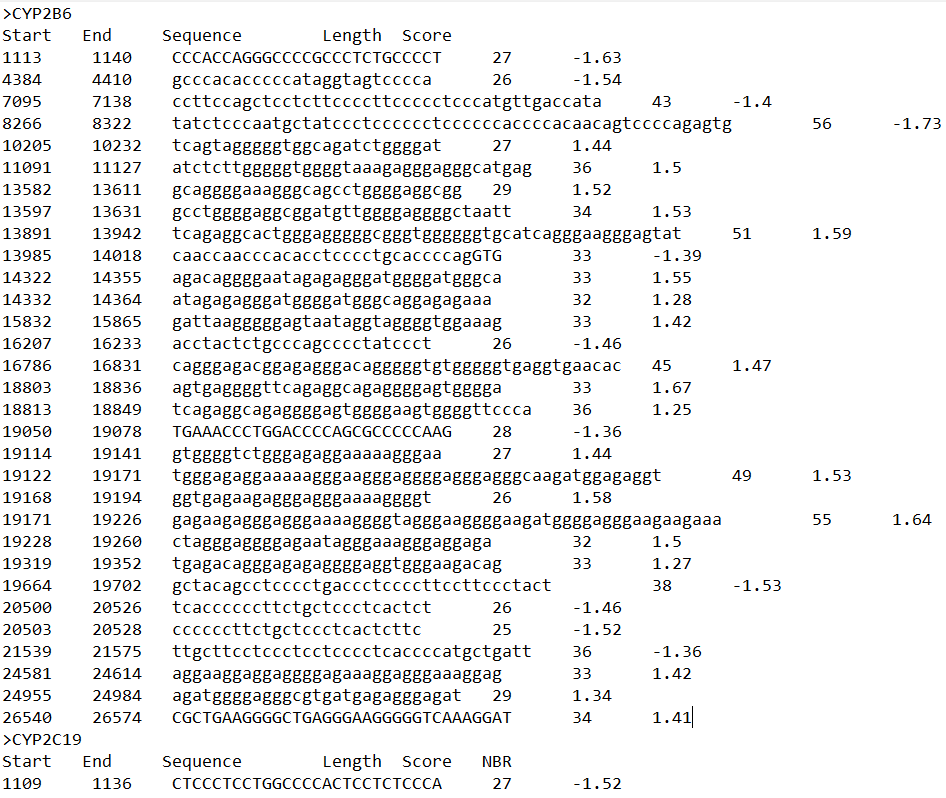
1. **Procedure**

Python program version 3.9.7 or later is preferred to be installed along with the libraries Biopython, Matplotlib, and NumPy to run the code. This work expands the functionality of our previous two codes “SNP-locator” and “G4-overlap” [7] and eliminates the need for a user manual.

**2.1 Workflow**

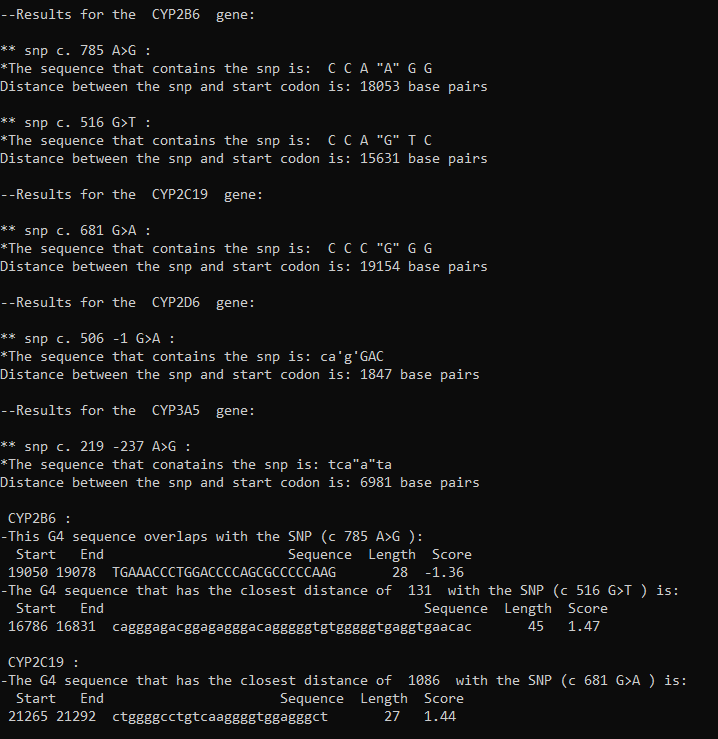
The developed code first locates the distances of SNPs relative to the start codon and their respective positions (flanking region) in the genomic file. This is achieved by calling a function that opens the input directory provided by the user to read each Fasta file individually. Moreover, as the files are being read, the start codon is detected, and the number of base pairs in the exon region is tracked. Once this number reaches the SNP position in the coding DNA (c.DNA), the function checks if there is a particular distance upstream or downstream of the reached position that needs to be traveled and returns the total distance of the SNP accordingly. The function repeats this step for all the SNPs of the genes available and stores the results in a dictionary.

Then, since the code incorporates the G4Hunter tool with some modifications to allow it to run on multiple Fasta files, a new text file called “-G4\_Merged.txt” will be automatically created in the directory provided by the user, containing the predicted G4 sequences for the genes with their start position, end position, length, and scores (Figure 1).



**Figure 1.** The G4 sequences predicted by the G4Hunter tool with their positions (start and end), length, and scores

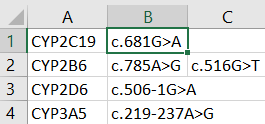
Another function in the code will store the locations of G4 sequences in a dictionary while reading the “-G4\_Merged.txt” file and performs two consecutive computations to map the sequences with the SNPs. First, it calculates the distance between G4 sequences and the start codon. Afterward, it computes the distance between each G4 sequence and the SNPs, by either subtracting the end position of the G4 sequence from the SNP position if the SNP is found downstream of the G4 sequence or by subtracting the SNP position from the start position of the G4 sequence if the SNP is found upstream of the G4 sequence. The same process is repeated for every provided gene and the results are stored in a list. Finally, the SNPs’ positions in the genomic file and the G4 sequences, along with their characteristics, that have the lowest distance from the SNPs are returned in the output as shown in Figure 2.



**Figure 2.** Representation of the output of the code

**2.2 Data Entry**

In order for the code to execute properly there exist some guidelines on the input data. Fasta files with 1000 nucleobase pairs before the promoter are accepted, however, they have to include capital letters for bases in the exon regions and small letters for bases in the intron regions. Such files can be obtained from the UCSC Genome Browser, where each file represents one whole gene. Furthermore, the code also takes the path of the CSV file that contains the SNPs of the genes in the format as shown in Figure 3. It is important that the locations of the selected SNPs are in the cDNA and that all of the Fasta files are in the same directory.



**Figure 3.** Format of the CSV file with the SNPs of the genes.

**2.3 Instructions to run the code**

To run the code, the terminal has to be opened in the same directory as that of the code. Once that is confirmed, the following command can be passed: ("python <code\_name.py> -i <inputrepository> -o <outputrepository> -w <window size> -s <score threshold>"). Then, the user will be prompted to enter the direct path to the CSV file. After entering the CSV file path, the code will be executed, generating a file that contains the predicted G4 sequences in the output repository, and displaying the closest G4 sequences from the SNPs on the terminal. Following these steps and the guidelines for the input data errors are avoided.

**3. Summary and Validation**

In summary, this code was developed to contribute to the studies that focus on discovering the functionality of the G4 quadruplexes. In particular, it detects G4 sequences that lie close to SNPs found in genes and allows future studies to analyze specific G4 sequences for gene expression and diseases, instead of examining a range of possible G4 sequences that could be time-consuming. Moreover, with this code, it is possible to detect SNPs that overlap with the G4 sequences which might be a potential factor for the formation of the latter. Furthermore, the code was tested on the same set of CYP genes used in our previous study [7], and the obtained results were identical.

**Availability and implementation**

The code can be found at https://github.com/Marc-shebaby/Capstone-Project.git.

1. **References**

1. Baral, A., Kumar, P., Halder, R., Mani, P., Yadav, V. K., Singh, A., Das, S. K., & Chowdhury, S. (2012). Quadruplex-single nucleotide polymorphisms (Quad-SNP) influence gene expression difference among individuals. Nucleic acids research, 40(9), 3800–3811.

2. Bedrat, A., Lacroix, L., & Mergny, J. L. (2016). Re-evaluation of G-quadruplex propensity with G4Hunter. Nucleic acids research, 44(4), 1746–1759.

3. Brázda, V., Luo, Y., Bartas, M., Kaura, P., Porubiaková, O., Šťastný, J., Pečinka, P., Verga, D., Da Cunha, V., Takahashi, T. S., Forterre, P., Myllykallio, H., Fojta, M., & Mergny, J. L. (2020). G-Quadruplexes in the Archaea Domain. Biomolecules, 10(9), 1349.

4. Cantara, A., Luo, Y., Dobrovolná, M., Bohalova, N., Fojta, M., Verga, D., Guittat, L., Cucchiarini, A., Savrimoutou, S., Häberli, C., Guillon, J., Keiser, J., Brázda, V., & Mergny, J. L. (2022). G-quadruplexes in helminth parasites. Nucleic acids research, 50(5), 2719–2735.

5. Davis J. T. (2004). G-quartets 40 years later: from 5'-GMP to molecular biology and supramolecular chemistry. Angewandte Chemie (International ed. in English), 43(6), 668–698.

6. Löwik, M. M., Groenen, P. J., Levtchenko, E. N., Monnens, L. A., & van den Heuvel, L. P. (2009). Molecular genetic analysis of podocyte genes in focal segmental glomerulosclerosis--a review. *European journal of pediatrics*, *168*(11), 1291–1304.

7. Saad, M., Shebaby, M., Mehawej, C., & Faour, W. (2022). New complementary python codes to locate Single Nucleotide Polymorphisms (SNPs) and Overlapping G-Quadruplex Sequences (G4s). MethodsX, 9, 101875.

7. Sen, D., & Gilbert, W. (1988). Formation of parallel four-stranded complexes by guanine-rich motifs in DNA and its implications for meiosis. *Nature*, *334*(6180), 364–366.

8. Sim, S.C., Ingelman-Sundberg, M. The Human Cytochrome P450 (CYP) Allele Nomenclature website: a peer-reviewed database of CYP variants and their associated effects. Hum Genomics 4, 278 (2010).

9. Zeraati, M., Moye, A. L., Wong, J. W. H., Perera, D., Cowley, M. J., Christ, D. U., Bryan, T. M., & Dinger, M. E. (2017). Cancer-associated noncoding mutations affect RNA G-quadruplex-mediated regulation of gene expression. Scientific reports, 7(1), 708. (10)