```
function sequence_FT = get_sequence_ft(hbb, sequence_number, N)

        sequence = hbb.Sequence;
        cds_indices = hbb.CDS.indices;

        % Indices
        % 1: 27726
        % 2: 27817
        % 3: 27940
        % 4: 28162
        % 5: 29019
        % 6: 29147

        % indices for coding & non-coding regions
        % non-coding regions are sequence[1 2]a
        sequence1 = sequence(cds_indices(1) : cds_indices(2));
        sequence1a = sequence(cds_indices(2) : cds_indices(3));

        sequence2 = sequence(cds_indices(3) : cds_indices(4));
        sequence2a = sequence(cds_indices(4) : cds_indices(5));

        sequence3 = sequence(cds_indices(5) : cds_indices(6));

        % convert A, T, C, G to binary numbers
        switch sequence_number
                case 1
                        coding_A = (upper(sequence1) == 'A');
                        coding_T = (upper(sequence1) == 'T');
                        coding_C = (upper(sequence1) == 'C');
                        coding_G = (upper(sequence1) == 'G');
                case '1a'
                        coding_A = (upper(sequence1a) == 'A');
                        coding_T = (upper(sequence1a) == 'T');
                        coding_C = (upper(sequence1a) == 'C');
                        coding_G = (upper(sequence1a) == 'G');
                case 2
                        coding_A = (upper(sequence2) == 'A');
                        coding_T = (upper(sequence2) == 'T');
                        coding_C = (upper(sequence2) == 'C');
                        coding_G = (upper(sequence2) == 'G');
                case '2a'
                        coding_A = (upper(sequence2a) == 'A');
                        coding_T = (upper(sequence2a) == 'T');
                        coding_C = (upper(sequence2a) == 'C');
                        coding_G = (upper(sequence2a) == 'G');
                case 3
                        coding_A = (upper(sequence3) == 'A');
                        coding_T = (upper(sequence3) == 'T');
                        coding_C = (upper(sequence3) == 'C');
                        coding_G = (upper(sequence3) == 'G');
                case 4
                        coding_A = (upper(sequence) == 'A');
                        coding_T = (upper(sequence) == 'T');
                        coding_C = (upper(sequence) == 'C');
                        coding_G = (upper(sequence) == 'G');
```

```
        end

        % get FFT of sequence by summing up FFT of each sequence of binary
        % numbers
        sequence_FT = abs(fft(coding_A, N)) + ...
                abs(fft(coding_T, N)) + ...
                abs(fft(coding_C, N)) + ...
                abs(fft(coding_G, N));

        % sequence_FT = abs(fftshift(fft(coding_A, N))) + ...
        %     abs(fftshift(fft(coding_T, N))) + ...
        %     abs(fftshift(fft(coding_C, N))) + ...
        %     abs(fftshift(fft(coding_G, N)));

        sequence_FT = sequence_FT(15:length(sequence_FT)-15);    % remove spikes @ beginning &
 end

end
```

```
Not enough input arguments.

Error in get_sequence_ft (line 3)
        sequence = hbb.Sequence;
```

Published with MATLAB® R2018b