GENETICS, SOFTWARE

PARALLELIZED COMPUTER CODE AND DNA TRANSCRIPTION

31.01.2024

At stackexchange there is a super interesting discussion on parallelized computer code and DNA transcription (which is different to the DNA-based molecular programming literature…)

> IF : Transcriptional activator; when present a gene will be transcribed. In general there is no termination of events unless the signal is gone; the program ends only with the death of the cell. So the IF statement is always a part of a loop.
>
> WHILE : Transcriptional repressor; gene will be transcribed until repressor is not present.
>
> FUNCTION: There are no equivalents of function calls. All events happen is the same space and there is always a likelihood of interference. One can argue that organelles can act as a compartment that may have a function like properties but they are highly complex and are not just some kind of input-output devices.
>
> GOTO is always dependent on a condition. This can happen in case of certain network connections such as feedforward loops and branched pathways. For example if there is a signalling pathway like this: A → B → C and there is another connection D → C then if somehow D is activated it will directly affect C, making A and B dispensable.

Of course these are completely different concepts. I fully agree with the further stackexchange discussion that

> it is the underlying logic that is important and not the statement construct itself and these examples should not be taken as absolute analogies. It is also to be noted that DNA is just a set of instructions and not really a fully functional entity … However, even being just a code it is comparable to a HLL [high level language] code that has to be compiled to execute its functions. See this post too.

Please forget everything you read from Francis Collins about this.