Multisubjective HCR nucleic acid design assistant

v1.0-b1/2011-09-26

by John P. Sadowski

Overview

Multisubjective improves the quality of sequence designs for nucleic acid hairpin chain reaction (HCR) systems, and automates several tasks in the process of sequence design. Multisubjective works in concert with another designer, such as the NUPACK multiobjective designer or Dave Zhang's Domain Design (DD), to minimize undesired secondary structure in these systems.

It has been observed that the NUPACK multiobjective designer does a poor job of avoiding undesired secondary structure in regions intended to be unpaired, when used to design larger systems (~9 hairpins). Multisubjective identifies a minimal set of specific bases which are responsible for undesired secondary structure, and instructs the designer to redesign only those bases, holding the others constant. Multisubjective also enforces exclusion of sequences consisting of a certain number of a single base type in a row, including mixed bases such as S or W.

Multisubjective also streamlines the design process by interacting directly with the NUPACK web server's multiobjective design software. Multisubjective is capable of automatically downloading and processing NUPACK job results, and of automatically submitting new design jobs to the web server. Multisubjective is also capable of interaction with DD, as it can read and write files in the DD format.

Multisubjective requires a local copy of NUPACK 3.0 for the secondary structure analysis, and cURL 7.21.6 is required for automatic interaction with the NUPACK web server.

Input

Multisubjective requires three input files, which should be placed in the Multisubjective working directory, which can be specified within the program interface.

mo.txt: NUPACK multiobjective data

The file *mo.txt* contains the NUPACK multiobjective design specification for the structure, with the "sequence=" group of lines removed and replaced by a single grave character (`). The grave character tells Multisubjective where to insert the new sequences after they have been generated.

ms.txt: Multisubjective data

The file *ms.txt* contains data needed by Multisubjective that is absent in the NUPACK specification. Most of these data relate to the function of the strands within the HCR system. The

syntax is identical to the NUPACK syntax with different keywords. The information in the file is as follows.

A hairpin is specified using the syntax "hairpin A1 : +" where A1 is the name of the strand (which must correspond to the name used in the NUPACK specification), and one of the characters '+' or '-' to specify the polarity of the hairpin. A hairpin has positive polarity if its toehold is at the 5' end of the strand, and has a negative polarity if its toehold is at the 3' end.

Including a bridge complex as part of the NUPACK design specification is often useful when bridge domains are present in the system. For each bridge interaction, a structure is specified containing the open portion (i.e., the sequence with the input domain removed) of the two strands involved in the bridge. Bridge complexes are specified using the syntax "bridge B1 C2 : br12" where B1 and C2 are the names of the two individual strands involved in the bridge, and br12 is the name of the complex containing both open portions. Again, all three of these names must appear as structures in the NUPACK specification. Currently, bridge complexes are expected to be open bridges with the positive-polarity strand positioned first.

Support for other classes of structures, such as cooperative-assembly strands, double hairpins, and closed bridges may be added in the near future.

Lastly, Multisubjective requires information about the length of each block and the location of bases desired to be immutable. The syntax of this group of lines is almost identical to the "sequence=" lines omitted from the NUPACK specification, but with the keyword "length". Instead of specifying bases, the letter 'N' represents a normal base and the letter 'Y' represents an immutable base. For example, "length a = 5N2Y" indicates that the length of block a is 7 nucleotides, and the last two bases are immutable. Specifying immutable bases is useful in the case of endonuclease restriction sites, or for pre-specifying sequences of clamping regions.

sequences.ms or sequences-dd.ms: Strand or block sequences

Multisubjective loads the actual sequences in one of two formats. The file *sequences.ms* is automatically obtained from the NUPACK web server. (Note that Multisubjective will overwrite this file if a new job is requested.) The file *sequences-dd.ms* is used if Multisubjective is told to load data in DD format.

Output

Multisubjective outputs data into three files, with an extra two generated if autosubmission to the NUPACK web server is utilized. These are all placed into the Multisubjective working directory specified in the program interface.

output.ms and output-dd.ms: New block sequences

The file *output.ms* countains the sequences of all the blocks, including mixed bases as determined by the Multisubjective algorithm, in the NUPACK multiobjective "sequence=" format. These can be copy-pasted into a NUPACK multiobjective design specification for manual input to the web server. The file *output-dd.ms* contains the same sequences in DD format, but with each mixed base set to a random base comprising that mixed base, and with all the non-mixed bases set to be immutable by DD.

output-post.ms and response.ms: NUPACK web server autosubmission data

If automatic submission to the NUPACK web server is utilized, Multisubjective will generate two additional files. The file *output-post.ms* contains a machine-readable version of the full NUPACK multiobjective design file including the new block sequences with mixed bases, used in the HTTP POST request to the NUPACK webserver. The file *response.ms* contains the web server's response to the POST request. If successful, the file contains the URL of the newly created job, and a subsequent run of Multisubjective can use this file to autofill the job number and token, to simplify the design process.

log.ms: Internal data log

The file *log.ms* contains a mirror of internal data stored after Multisubjective has finished loading and processing the three input files. It is useful for debugging purposes.

Program operation

Upon starting, the program gives the user a choice of input modes:

- Input by job number. If the user inputs a number, the program checks whether the job has already been downloaded within the current working directory. If not, the program asks for a token and trial id, uses cURL to download the job from the NUPACK web server to the file *mo-output.zip*, and unzips the job into the folder */mo_output* . It then copies the requested sequence file to *sequences.ms*, overwriting a previous file if necessary. If the job has already been downloaded, it asks only for a trial id, and copies the desired file from */mo_output* to *sequences.ms* .
- Run last job. If the user inputs the letter 'l', the preexisting *sequences.ms* file is used. This is useful for running the same sequence many times.
- Autofill job information. If the user inputs the letter 'a', then the program gets the job number and token from *response.ms*, which contains the information for the job most recently submitted to the NUPACK web server, in a previous run of Multisubjective. This is useful when iterating sequences between Multisubjective and NUPACK multiobjective design.
- Load DD file. If the user inputs the letter 'd', the program loads sequence data from the file *sequences-dd.ms* . Multisubjective automatically converts the block sequences contained in the DD file into the strand sequences used by Multisubjective.

- Set other options. If the user inputs the letter 's', the program prompts the user for the Multisubjective working directory (where the input and output files are stored), the local NUPACK home directory, and the email address used during autosubmission to the NUPACK web server. These are stored in the file *multisubjective-settings.ms* in the operating system's default working directory, usually the user's home directory and automatically reloaded on subsequent runs of Multisubjective.

After data has been loaded from the input files, Multisubjective uses the local installation of NUPACK to analyze the secondary structure of the strands. This is done in two passes. The first pass contains the full sequence of each hairpin, testing the "closed" hairpin structures. The second pass includes both the hairpins with their input domains removed, giving the "open" hairpin structures, as well as the bridge complexes. All NUPACK analysis files are stored in the directory */nupack* .

Multisubjective then tabulates the undesired secondary structure from the NUPACK analysis, and decides which bases need to be changed to disrupt this secondary structure, avoiding changing the desired bases in the hairpin and bridge complexes, as well as the immutable bases calculated from user input. All such bases are changed to N. It also checks for the existence of prevented sequences, and inserts the proper mixed base to disrupt this sequence (e.g., if there are too many A's in a row, one of them will be changed to a B; if there are too many W's in a row, one will be changed to an S), again avoiding changing immutable bases. The algorithm seeks to minimize the number of changed bases by changing an N to the mixed base if one is in the proper range of positions. If the presence of immutable bases prevents any of these changes from being made, a warning is output to the screen.

Once these changes have been made to the strand sequences, the block sequences must be extracted for output. Since each block may appear many times throughout the system, and each instance may have been changed in different ways, a base collision scheme is used to resolve these differences. Mixed bases other than N take precedence over N's, N's take precedence over single bases, and for two non-N bases, the intersection of the two bases is taken unless the intersection is empty, in which case the earlier base is used and a warning is output to the screen.

Once the strand sequences have been compiled, they are written to the output files and the user is asked whether submission to the NUPACK web server is desired. If so, Multisubjective generates a formatted version of the NUPACK multiobjective design file with the new sequences, and uses cURL to submit an HTTP POST request to the NUPACK web server. Multisubjective then cheerfully exits.