

## Quick start guide

### Overview

Multisubjective is a nucleic acid design tool that identifies and removes strong spurious base pairing in arbitrary user-defined systems, with special features targeted towards dynamic strand-displacement systems used in developmental self-assembly. Multisubjective identifies the small subset of bases responsible for the spurious base pairing, and uses a client designer to select new identities for those bases.

Multisubjective's algorithm is based on multiple iterations of an analysis–redesign cycle with the goal of finding a design that has very few strong undesired base pairs. Multisubjective uses a thermodynamic analysis to identify a minimal set of bases that need to be changed because they are responsible for undesired secondary structure, or because they are part of a user-defined prevented sequence. This analysis is passed to the client designer to redesign only those bases identified as problematic by Multisubjective, holding the others constant. Multisubjective interacts automatically with an implementation of the combinatorial heuristics-based Domain Design (DD) software,<sup>1</sup> and is also capable of automatically submitting a design to the NUPACK multi-objective designer<sup>2,3</sup> on their web server<sup>4,5</sup> and automatically retrieving the results.

There are two ways to use Multisubjective. A standalone text-based interface can take input either interactively from the user or as part of a command line call. Multisubjective has also been integrated into DyNAMiC Workbench, an integrated development environment for motif and sequence design of developmental self-assembly systems, providing a convenient front-end graphical user interface for Multisubjective.<sup>6,7</sup>

---

<sup>1</sup> D. Y. Zhang. Towards domain-based sequence design for DNA strand displacement reactions. *Lect. Notes Comp. Sci.*, 6518:162–75, 2011. doi:10.1007/978-3-642-18305-8\_15

<sup>2</sup> J. N. Zadeh. Algorithms for nucleic acid sequence design Ph.D. dissertation, California Institute of Technology, 2010. Chapter 3: Sequence design for multi-state nucleic acid systems. <http://thesis.library.caltech.edu/5801/>.

<sup>3</sup> B. R. Wolfe and N. A. Pierce. Sequence design for a test tube of interacting nucleic acid strands. *ACS Synth. Biol.* In press, 2014. doi:10.1021/sb5002196

<sup>4</sup> J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, and N. A. Pierce. NUPACK: analysis and design of nucleic acid systems. *J. Comput. Chem.*, 32:170–3, 2011. doi:10.1002/jcc.21596

<sup>5</sup> NUPACK. <http://nupack.org/>. Accessed October 28, 2014.

<sup>6</sup> C. Grun, J. Werfel, D. Y. Zhang, and P. Yin. DyNAMiC Workbench: an integrated development environment for dynamic DNA nanotechnology. Manuscript in preparation.

<sup>7</sup> DyNAMiC Workbench. <http://molecular-systems.net/workbench/>. Accessed October 28, 2014.

## ***Program operation***

Multisubjective requires a specification file that is an augmented version of the NUPACK “.np” format.<sup>8</sup> (Workbench uses the extension “.ms” instead of “.np” for these files.) The format of the specification file is explained in the full documentation.

Sequences can be loaded from the specification file itself, or from a separate file or set of files of either the DD “.dd” or NUPACK “.npo” formats. The available sequence input modes are:

- Fill with random bases (f). The program creates ten sets of random block sequences consistent with the base constraints in the specification file, and uses these to construct ten sets of strand sequences. This option should be used if sequences are to be directly loaded from the specification file, even if none of them are degenerate. (Note that the constraints will not be honored in subsequent rounds unless they are set as immutable bases.)
- Seed with independent DD trials (i). DD is run ten times from ten independent seeds consistent with the base constraints in the specification file, and the resulting ten sets of sequences are used as the initial candidate sequences. (Note that the constraints may not be honored by DD and will not be honored by Multisubjective in subsequent rounds unless they are set as immutable bases.)
- Load a DD file (d). The program loads sequence data from the file *candidate.dd* . Multisubjective automatically converts the block sequences contained in the DD file into the strand sequences used by Multisubjective.
- Load multiple DD files (m). The analysis will run ten times using the files *candidate-1.dd* to *candidate-10.dd* . This option is useful for quickly comparing several related DD outputs and determining which is optimal.
- Load a NUPACK-MO file (n). The program loads sequence data from the file *candidate.npo* .
- Autofill from last MO web submission (a). The program uses the job number and token from the job most recently submitted by Multisubjective to the NUPACK web server. This is useful when iterating sequences between Multisubjective and NUPACK multi-objective design.
- Input by job number (j). If the user inputs a number, the program asks for a token, and uses cURL to download the job from the NUPACK web server. It then asks for the trial id, or the keyword “all” can be used to test trials 0–9 in that job. If the job has already been downloaded, it asks only for a trial id, and uses the already-downloaded file.

---

<sup>8</sup> NUPACK design demos. <http://nupack.org/design/demos>. Accessed October 28, 2014.

The user also has a choice of iteration modes to specify what designer is to be used to select new identities for those bases that were identified by Multisubjective as problematic. If multiple sequences are tested by Multisubjective, only the one with the fewest number of undesired bases is passed to the designer. The iteration modes are:

- Run DD once (o). The program runs one round of DD, yielding ten new designs.
- Run DD in loop (l). Ten rounds of DD are run, with ten new designs generated in each round with only the one with the fewest number of undesired bases advancing to the next round.
- Submit to NUPACK-MO web server (w). Multisubjective generates a formatted version of the NUPACK multi-objective design file with the new sequence assignments inserted. Multisubjective then uses cURL to submit an HTTP POST request to the NUPACK web server.
- Use random bases in loop (r). Ten rounds are run, with ten new random designs generated in each round (within the prevented sequence constraints) with only the one with the fewest number of undesired bases advancing to the next round.
- No designer (x). No designer is used after Multisubjective analyzes the input design for undesired secondary structure.

### ***Using the Workbench interface***

Multisubjective is one of many tools available on DyNAMiC Workbench. The web application is available at <http://yin.hms.harvard.edu/workbench>. As Workbench is currently in limited alpha release, users must request access in order to create an account.

A structural specification can be created in two ways. First, an external “.np” file can be uploaded to the server by dragging into the “Files” pane on the right. (If it gets stuck on the “loading” icon, reload the page to update the display. If the browser displays the file instead of uploading it, create a new “.np” file and paste the contents of the local file into it.) The file must then be renamed with a “.ms” extension for Workbench to recognize it as a Multisubjective task.

Alternatively, structural specifications can be constructed within the IDE. On the dashboard tab, under the “Domain-level systems” menu select “Domain-level System Editor (DIL)”. Systems are built up in three stages: sequence segments, strands, and complexes.

- The first step is to add sequence segments on the right pane, and adjust their length and optionally their sequences. For example, clicking the “Add” button four times will create four segments named “1”, “2”, “3”, and “4”.
- Second, add strands on the bottom pane, and under the “Segments” column type the names of the segments that make up the strand, separated by spaces, with an asterisk following a name representing a complement. For example, clicking the “Add” button once will create a strand named “n1”; a simple hairpin can be represented as “1 2 3 2\* 4”.

- Lastly, add complexes in the central plane. This opens a dialog box; in the “Strand Order” field enter the strand name(s) to be included, and in the “Structure” box enter the complex’s structure in dot-parenthesis notation by segments. For example, enter “n1” and “.(.).” respectively to create a hairpin structure.

More documentation of the DIL interface is available by clicking “Help” at the very top of the screen and selecting “DyNAMiC Intermediate Language”. When finished, click “Multisubjective” on the tab’s top panel to send it to the Multisubjective interface.

It is useful to place the “.ms” file in its own folder due to the large number of files generated by the design process. To do this, right-click your email address at the top of the “Files” pane, mouse-over the “Create” option, enter a filename, mouse-over the second “Create” option, and click “Folder” at the bottom of the menu. Then drag the “.ms” file into the folder.

Multisubjective’s options can be set through the toolbar at the top. The input and iteration modes can be set in the menus at the top of the tab. Note that not all input and iteration modes are operational through the Workbench interface. The “Advanced options” button brings up a dialog box that can be used to edit the thresholds, the intermolecular and immutable modes, the functional identities of the complexes, and the thermodynamic settings. Clicking “Run” performs the design; as this takes several seconds or minutes, it can be monitored through the “Tasks” tab (available under the “System” menu at the top left of the screen), or by clicking the very small arrow at the center bottom of the screen to reveal the console display.

As an example, choose the “Fill with bases from this design view” and “Run DD once” modes and click “Run”. Multisubjective creates random sequences and analyzes them, displaying the results for the best of the ten trials: strong spurious base pairs are connected by red lines, and prevented sequences are highlighted in orange. After the design run, ten files *candidate-1.dd* through *candidate-10.dd* have been created, which are the results of running DD on the analyzed sequences. (Note that the analysis displayed is always of the input sequence for that particular run, not the newly generated sequences.) Further iterations can be run by changing the input mode to “Load sequences from multiple DD files”; each time “Run” is clicked, the old candidate sequences are loaded, analyzed, and replaced with a new set of sequences.

### ***Using the console interface***

To install Multisubjective locally:

- Install NUPACK 3.0.4 (<http://www.nupack.org/>).
- Obtain the file multisubjective.cpp (v1.1.2) and compile it using Xcode, gpp, or another compliant compiler.

If automatic interaction with the NUPACK web server is to be used:

- Install cURL 7.21.6 (<http://curl.haxx.se/>).

If automatic interaction with DD is to be used:

- Install Node.js 0.6.12 (<http://nodejs.org/>).
- Create the directory \$HOME/Documents/Multisubjective , and run the command “npm install commander underscore format” in that directory.

- Obtain the files `dd.js` (v0.3) and `cldd.js` (v0.4.7.3) and place them in that directory. Set `cldd.js` to be executable.

Running Multisubjective without any command line arguments brings up a menu where the user can select the input and iteration modes. There is also a “Set other options” submenu that allows the user to change the Multisubjective working directory (where the input and output files are stored), the local NUPACK home directory, the specification filename, and the filename prefixes of the sequence input and output files.

Alternatively, the following command-line options can be used to run Multisubjective non-interactively:

```
multisubjective -m mode [-j job_number] [-r trial_id] [-k token] [-c configuration_filename]  
[-d working_directory] [-i specification_file_prefix] [-s sequence_file_prefix] [-o output_file_prefix]  
[-h NUPACK_home_directory]
```

Here, “mode” is a two-character string representing the letters for the input and iteration modes; this is the only required argument. The rest of the arguments are largely self-explanatory: `-j`, `-r`, and `-k` are for getting files from the NUPACK web server, and the others are various filenames. More information is available in the full documentation.