The point

The following describes a method to design simple, orthogonal DNA sequences from scratch. The primary failure modes are:
1. long, unintended complementary domains
2. stable secondary structure

The basic idea

Some 25 years ago, Ned Seeman devised a method ("*De Novo* design of sequences for nucleic acid structural engineering," Journal of Biomolecular Structure and Dynamics, 8, 1990) for designing single-stranded DNA sequences that minimizes repeating domains and stable secondary structure. The idea is to choose a cutoff domain length (e.g. 3 bases), construct a library of unique sequences of this length (e.g. there are 64 unique 3-base domains), and then use the library as a basis to construct the sequence one base at a time.

The method

1. Choose a 3-base codon at random and write it down. This is the beginning of your growing sequence.
2. Remove the chosen codon and its complement from the library.
3. Choose a 3-base codon whose first 2 bases are the same as the last 2 bases of the previous codon (also at random).
4. Add the 3rd base of the chosen codon to the end of your growing sequence.
5. Repeat steps 2-4 until your sequence is the desired length.

An example

aag	tgc	cta	aga	gtt	ttg
act	gca	ttc	atg	gta	tac
cct	ggc	tgg	ctt	ata	gcc
cag	aca	gaa	ctg	atc	gcg
tta	acg	gct	tat	cca	acc
tcc	cgg	ggt	tag	ctc	agc
cga	tct	tca	gac	gtc	ccg
aac	agt	cgc	tcg	gtg	agg
att	cac	tgt	gat	aat	caa
cat	cgt	taa	tga	gag	gga

5′- A C G A T ... - 3′

The logic

Because each 3-base codon can be used at most one time, unintended complementary domains between sequences can be no longer than 2 bases. Because the complement to each 3-base codon can be used at most one time, complementary domains within a sequence can be no longer than 2 bases, thus destabilizing secondary structures.

Other thoughts

In practice, I usually exclude 'AAA', 'TTT', 'GGG', and 'CCC' from the library. 'AAA' and 'TTT' conflict with the polyT spacers that we typically use; 'CCC' and 'GGG' stabilize hairpins and can hybridize in non-Watson-Crick ways (e.g. G-quadruplex).

If your application is particularly sensitive to the presence of stable hairpins, I would suggest using a library made of only: (1) 'A's, 'T's, and 'G's; or (2) 'A's, 'T's, and 'C's. Although this significantly reduces the number of distinct sequence possibilities, it also prevents you from designing sequences that contain both 'C's and 'G's (the base pairs that typically stabilize hairpin stems).

Screening designed sequences

The above procedure is not foolproof. Fortunately, there are simple web tools that can help to double-check your sequences.

NUPACK

NUPACK is a software suite for the design and analysis of nucleic acid systems developed by Niles Pierce at Caltech. We use it primarily to check the temperature dependence of hybridization between a number of interacting DNA sequences. It is also useful for confirming that supposedly non-complementary sequences are indeed non-complementary, although it can do quite a bit more.

mfold

mfold is a web server developed by Michael Zuker at RPI. We use it primarily to look for stable secondary structures in single-stranded DNA sequences. Like NUPACK, it also has many other features that might be useful for specific applications. For example, DINAMelt is a favorite of certain theoretical groups (Frenkel, etc.) for estimating hybridization free energies.

Written by my friend Ben Rogers at Brandeis University