# The TRC shRNA Design Process

## Overview

- We design shRNA constructs ("clones") with an algorithm. Our algorithm uses several criteria to rank potential 21mer targets within each human and mouse Refseq transcript. The algorithm applies a set of rules, including those derived from the siRNA literature, analysis of TRC library performance datasets, constraints on the synthesis and cloning of the oligonucleotides and others. In applying the algorithm, our aim is to achieve a balance of two competing goals: make hairpins that effectively knockdown the target transcript and, as best possible, design hairpins that knockdown only one gene and do not directly alter other genes (so-called 'off-target' effects). Each goal presents distinct challenges. The criteria for predicting effective knockdown with either siRNA or shRNA are not well understood and are still being developed and refined. Specificity is constrained by genome evolution--since many genes are part of extensive gene families, targeting a specific family member can be difficult. Furthermore, functionally distinct genes share many motifs with underlying nucleic acid sequence similarity. Our knowledge of transcript structure and variants is still very incomplete as well. For all these reasons and more, we construct several shRNAs for each transcript with the expectation of getting a range of knockdown efficiencies across the set and at least a few which knockdown effectively.
- Users of this database should be aware that in order to have consistent and reliable annotation, the RNAi Consortium decided early on to use NCBI's REFSEQ collection of transcripts as the definitive source of i nformation for the primary target sequence for the design of shRNAs.
- As a general rule in the construction of the library, we construct shRNAs targeting just one Refseq transcript for each NCBI gene. because of the high sequence identity among different transcripts from the sa me gene, the majority of the shRNAs target all known transcript variants.

## A brief narrative of the candidate selection process

- **Get the Candidate Sequences**

  For one representative human and mouse Refseq transcript per gene, we assess all 21mers starting 25 bp after the beginning of the CDS up to those starting 150 bp from the end of the transcript. Each 21mer is called a 'candidate'.
- **Score the Candidate Sequences For Knockdown Efficiency**

  Each candidate is given an "original score" by applying a set of rules that either penalize or reward features predicting successful knockdown and clone-design considerations, and then calculating the product of all the penalties/rewards. The individual rules are listed below. Candidates are also rewarded or penalized based on the frequency of predicted microRNA-like off-target effects via "seed" matches. The candidates are then sorted by score and the best 250 continue to the next step.
- **Score the Candidates Sequences for Specificity**

  We calculate a "specificity factor" to promote candidates without significant sequence similarity to other genes. Each candidate is compared by BLASTN to two distinct abstractions of the transcriptome: the NCBI Unigene "unique" database (vaguely defined by NCBI as the "longest, best" sequence from each unigene cluster), and the transcripts from Refseq. Any candidate that has three or more differences (and at least two of the differences in the core positions 3-19, i.e., not on the ends of the 21mer target region) with every (non-self) gene is considered unique for that reference set. The "specificity factor" is highest when the candidate is unique in both the Unigene and Refseq datasets, and is lowest when it is not unique in both. The "specificity score" combines the "original score", the "miRNA seed frequency factor", and the "specificity factor".
- **Avoid overlapping shRNAs**

  In order to create a library of many distinct shRNAs for every human and mouse gene, we consider the target region of the target gene (e.g. CDS, 3'UTR) as well as overlap of top-scoring candidates with existing library clones and with each other. Candidates

are ranked by "specificity score" and then assessed for target region and overlap with other existing or ordered shRNAs until the desired number of candidates is selected per gene. We attempt to select new candidates that have a "specificify factor" and a "specificity score" greater than 1, no overlap with other TRC shRNAs, and are distributed in a 4:1 ratio between the CDS and 3'UTR. If sufficient candidates are not available that meet these criteria, we allow some (but not full) overlap and/or relax the CDS:3'UTR ratio before relaxing the score requirements.

## Current Rule Set

### Rule Set 9

| | Rule | Description |
|---|---|---|
| 1 | aaStart9 | Exclude any candidate beginning with AA (score = 0) |
| 2 | fourRow9 | Exclude any candidate containing a run of four of the same base in a row (score = 0) |
| 3 | gcScore9 | Exclude candidates with extreme GC percentage (GC <= 25% or > 60%); promote candidates with GC between 25-55% (score = 3); if GC > 55% and <= 60% then score = 1 (neutral) |
| 4 | nonGATC9 | Exclude any candidate containing ambiguous bases (e.g. N) (score = 0) |
| 5 | restrictionSite9 | Exclude any candidate containing certain restriction sites: ...GGTACC..., ...GAATTC..., ...CTCGAG..., ...CATATG..., ...ACTAGT..., ...GGTAC, ...GAATT, GTACC..., TACC..., CTAGT... |
| 6 | sevenGC9 | Exclude any candidate with a run of 7 C/G bases (score = 0) |
| 7 | stemLoopStem | Penalize candidates that can form an internal stem-loop (score = 0.1) (minimum stem length = 5, minimum loop size = 4) |
| 8 | threePrimeClamp6 | Give precedence to candidates with weaker base-pairing at positions 15-20 (priority on pos. 17-19); score = 5 if all 6 positions are A or T, decreasing to 0.1 if all 6 are G/C. Score drops off steeply as the number of A/T bases decreases. |

## Previous Rule Sets

### Rule Set 8

| | Rule | Description |
|---|---|---|
| 1 | aaStart | AAstart; candidates beginning with AA get a penalty of .000000000000001; |
| 2 | fourRow | fourInARow; any four of the same bases in a row gets the penalty of 0.01 |
| 3 | gcScore8 | gcContent: extremes of GC percentage are penalized; candidates with GC <= 25% or > 60% are penalized .01; with GC between 25-55% the candidate gets a reward of 3; with GC >550 and <=60% the score is 1 (neutral) |
| 4 | nonGATC | no ambiguous bases allowed in the candidate 21mer sequence |
| 5 | restrictionSite8 | GGTACC, GAATTC, CTCGAG, CATATG, ACTAGT, ...GGTAC, ...GAATT |
| 6 | sevenGC | sevenGC; any run of 7 C or G gets the penalty of 0.01 |
| 7 | stemLoopStem | Penalize candidates that can form an internal stem-loop (score = 0.1) (minimum stem length = 5, minimum loop size = 4) |

| | | |
|---|---|---|
| 8 | threePrimeClamp6 | Give precedence to candidates with weaker base-pairing at positions 15-20 (priority on pos. 17-19); score = 5 if all 6 positions are A or T, decreasing to 0.1 if all 6 are G/C. Score drops off steeply as the number of A/T bases decreases. |

## Rule Set 7

| | Rule | Description |
|---|---|---|
| 1 | aaStart | AAstart; candidates beginning with AA get a penalty of .000000000000001; |
| 2 | fivePrimeClamp | fivePrimeClamp:give precedence to a candidates with stronger base-pairing at the 5 prime end of the putative candidate, referred to as five_prime_clamp; penalty/reward .01 if first two positions are GG, .0001 if first two are TT; 2.5 if first four are (G\|C){4}; 2.4 if first three positions are G\|C{3}; 2.2 if begins (CC\|CG\|GC)(A\|T)(G\|C); 2 if begins (CC\|CG\|GC); 2 if begins (GC); 1.25 if begins (G\|C); 1 if begins (A\|T)(G\|C); .5 if begins ((A\|T){2} |
| 3 | fourRow | fourInARow; any four of the same bases in a row gets the penalty of 0.01 |
| 4 | gcScore | gcContent: extremes of GC percentage are penalized; candidates with GC \< 30% are penalized .01; with > 70% the penalty is .01; with GC between 30-50% the candidate gets a reward of 3; with GC >60 and \<70% the reward/penalty is 1 |
| 5 | internalAT | internalAT; we want to reward moderately AT rich regions from 7 through 10; if all four are A\|T, rewards is 2.2; if 3 of 4 are A\|T, the reward is 2, if 2 of 4 is A\|T, the reward is 1.5; if 1 or 4 is A\|T, the penalty is .7; if none of the four are A\|T, the penalty is 0.5 |
| 6 | internalATFlanking | internalATflank; we want to reward moderately AT-rich sequences at position 6 and 11; if both are AT, the reward is 1.2; if 1 is either A\|T, the reward is 1 and if neither is A\|T, the penalty is 0.85 |
| 7 | internalLoop | internalLoop: we penalize candidates that cand form a AAABBB loop with a 0.7 penalty |
| 8 | nonGATC | no ambiguous bases allowed in the candidate 21mer sequence |
| 9 | restrictionSite | GCCGGC, CCCGGG, CTCGAG, ...GCCGG |
| 10 | sevenGC | sevenGC; any run of 7 C or G gets the penalty of 0.01 |
| 11 | threePrimeClamp6 | Give precedence to candidates with weaker base-pairing at positions 15-20 (priority on pos. 17-19); score = 5 if all 6 positions are A or T, decreasing to 0.1 if all 6 are G/C. Score drops off steeply as the number of A/T bases decreases. |

## Rule Set 4

| | Rule | Description |
|---|---|---|
| 1 | aaStart | AAstart; candidates beginning with AA get a penalty of .000000000000001; |
| 2 | fivePrimeClamp | fivePrimeClamp:give precedence to a candidates with stronger base-pairing at the 5 prime end of the putative candidate, referred to as five_prime_clamp; penalty/reward .01 if first two positions are GG, .0001 if first two are TT; 2.5 if first four are (G\|C){4}; 2.4 if first three positions are G\|C{3}; 2.2 if begins (CC\|CG\|GC)(A\|T)(G\|C); 2 if begins (CC\|CG\|GC); 2 if begins (GC); 1.25 if begins (G\|C); 1 if begins (A\|T)(G\|C); .5 if begins ((A\|T){2} |
| 3 | fourRow | fourInARow; any four of the same bases in a row gets the penalty of 0.01 |
| 4 | gcScore | gcContent: extremes of GC percentage are penalized; candidates with GC \< 30% are penalized .01; with > 70% |

| | | |
|---|---|---|
| | | the penalty is .01; with GC between 30-50% the candidate gets a reward of 3; with GC >60 and \<70% the reward/penalty is 1 |
| 5 | internalAT | internalAT; we want to reward moderately AT rich regions from 7 through 10; if all four are A\|T, rewards is 2.2; if 3 of 4 are A\|T, the reward is 2, if 2 of 4 is A\|T, the reward is 1.5; if 1 or 4 is A\|T, the penalty is .7; if none of the four are A\|T, the penalty is 0.5 |
| 6 | internalATFlanking | internalATflank; we want to reward moderately AT-rich sequences at position 6 and 11; if both are AT, the reward is 1.2; if 1 is either A\|T, the reward is 1 and if neither is A\|T, the penalty is 0.85 |
| 7 | internalLoop | internalLoop: we penalize candidates that cand form a AAABBB loop with a 0.7 penalty |
| 8 | nonGATC | no ambiguous bases allowed in the candidate 21mer sequence |
| 9 | sevenGC | sevenGC; any run of 7 C or G gets the penalty of 0.01 |
| 10 | threePrimeClamp | threePrimeClamp: give precedence to a candidates with weaker base-pairing at the 3 prime end of the putative candidate; penalty/reward 5 if last three positions are A or T, 4.5 if last two are A\|T and third from is G\|C and fourth is A\|T; 4 if the last two are A\|T; 2 if the last base is A\|T; penalty is .2 if last two posisitions are G\|C; .5 if the last base is G\|C; 0.8 if the last base is G\|C and previous two are A\|T |