

1 **An unexpectedly effective Monte Carlo technique for the**
2 **RNA inverse folding problem**

3

4 Fernando Portela^{1*,2}

5

6 ¹ Citizen scientist, member of the Eterna Massive Open Laboratory

7 ² Department of Biochemistry, School of Medicine, Stanford University, CA 94305

8 * Corresponding author. Email: nando8888@gmail.com

9

10 **Abstract**

11 Solving the RNA inverse folding problem, also known as the RNA design problem, is critical to
12 advance several scientific fields like bioengineering, yet existing approaches have had limited
13 success. The problem has several features that resist traditional computational techniques, such
14 as its exponential complexity and the chaotic behavior of its cost function. Although some state-
15 of-the-art AI approaches have reported promising results, all existing computational methods
16 substantially underperform expert human designers. I combine a different technique, Nested
17 Monte Carlo Search (NMCS), with domain-specific knowledge to create an algorithm that
18 outperforms all prior published methods by wide margins and solves 95 of the 100 puzzles listed
19 in a recently proposed RNA solving difficulty benchmark.

20

21 **Keywords**

22 RNA design — RNA inverse folding — Nested Monte Carlo Search — Citizen Science

23

24 **Introduction**

25 The RNA inverse folding problem is crucial in numerous scientific fields such as pharmaceutical
26 research, synthetic biology and RNA nanostructures. Even though the question of the
27 computational complexity of the RNA design problem is not categorically settled, recent
28 evidence suggests it is NP-complete (Bonnet, Rządewski & Sikora, 2017). Moreover, target-
29 specific structural features like symmetries and short helices can heavily compound the difficulty
30 of solving a particular RNA design problem (Anderson-Lee et al., 2016).

31

32 It is therefore unsurprising that existing RNA design software packages explore search spaces by
33 way of trial and error. Examples of classic cost function minimization approaches include
34 packages like: RNAinverse (Hofacker, 2003), performing adaptive random walk; RNA-SSD
35 (Andronescu et al., 2004), using hierarchical structure decomposition; INFO-RNA (Busch &
36 Backofen, 2006), probabilistic sampling of sequences; NUPACK (Zadeh et al., 2011), ensemble
37 defect minimization; and MODENA (Taneda, 2015), a genetic algorithm. However, none of
38 these packages come close to matching the performance of talented human RNA designers in the
39 Eterna100 benchmark (Anderson-Lee et al., 2016): 54/100 for the best machine to 100/100 for
40 the most talented human experts. Recent efforts using more sophisticated AI techniques include
41 software packages like SentRNA (Shi, Das & Pande, 2018) (Eterna100 score 80/100) applying
42 Deep Learning techniques incorporating a prior of human design strategies, and MCTS-RNA
43 (Yang et al., 2017) (Eterna100 score 72/100) implementing a Monte Carlo Tree Search (MCTS)
44 process largely inspired by computational treatments of the game of Go (Gelly & Silver, 2008)
45 that were fashionable until the advent of DeepMind's alphaGo (Silver et al., 2017).

46
47 The particular form of MCTS implemented in MCTS-RNA, called Upper Confidence Bounds
48 applied to Trees (UCT) (Kocsis & Szepesvári, 2006), is known to be well suited for finding near-
49 optimal solutions in huge solution spaces, themselves embedded in typically gigantic search
50 spaces. Challenging RNA design problems often lack substantially large solution spaces though,
51 especially when considered in relation to the sizes of their respective search spaces. The rarity of
52 solutions within large subtrees effectively creates trap states (Ramanujan, Sabharwal & Selman,
53 2011) and causes the UCT search to ignore these subtrees for long periods of time (Mehat &
54 Cazenave, 2010), making UCT an ineffective approach in such contexts. However, the field of

55 General Game Playing (GGP) has produced more than one Monte Carlo algorithm. In particular,
56 the simpler and well-studied Nested Monte Carlo Search (Cazenave, 2009) algorithm has been
57 shown to be superior to UCT in many single player games (Mehat & Cazenave, 2010) and has
58 never been tried in the context of RNA inverse folding.

59
60 Meanwhile, even though SentRNA still underperforms human solvers in the Eterna100
61 benchmark, its incorporation of human design strategies—in the form of a dataset of experts’
62 solutions to numerous RNA puzzles—shows potential. However, the solution to a given puzzle
63 says nothing about *how* a human expert walked the tortuous path to the successful outcome. One
64 could conjecture that the process itself, not just the final product, probably holds consequential
65 and valuable information.

66
67 In addition, a quick survey of the newly collected move histories data on the Eterna game
68 platform (Lee et al., 2014) had convinced me that several behavioral patterns in players’ solving
69 styles could be encoded as algorithms. Combining all these observations, I hypothesized that
70 implementing a Nested Monte Carlo Search (NMCS) based RNA inverse folding agent enhanced
71 by heuristics in both the sampling and the explorative phases could lead to a best-of-class ability
72 to solve RNA design problems that are intractable to current computational methods.

73

74 **Methods**

75 The NESTed MOnTe Carlo RNA puzzle solver (NEMO) is implemented as a short single C++
76 file. The linked RNA folding engine is the ViennaRNA package (Lorenz et al., 2011) in its

77 version 2.1.9. A general overview of NMCS is presented in Fig. 1. NEMO's simple global
78 algorithmic layout is depicted in Fig. 2.

79

80 **Sampling phase heuristics**

81 The heuristics and strategies in the sampling phase—the playout policy in General Game Playing
82 theoretic parlance—are coded using domain knowledge acquired by personal experience.

83 Parameters affecting probabilities and distributions were chosen ad hoc, without performing
84 computational optimizations, whether gradient descent or otherwise. Its initial step consists in
85 filling up base pairs first, and only then the unpaired positions in the target structure. Following
86 this order allows NEMO to properly handle specific sub-goals like preventing unwanted base
87 pairings in 0-N bulges—a technique known as “blocking” among Eterna players—and using
88 thermodynamically favorable mismatches—also known as “boosting”—in multi-way junction
89 loops. (Note: In this paper, the term “mismatch” is meant to include all form of potential non-
90 canonical interactions at the end of helices, e.g. both terminal mismatches and dangling ends)

91

92 Roughly following the proportions found in known naturally occurring RNA structures
93 (Lemieux & Major, 2002), NEMO fills base pairs with a 60% GC, 33% AU and 7% GU
94 probability distribution, with a few exceptions for closing pairs of adjacent helices in junctions
95 and the closing/enclosing pairs of triloops. Unpaired bases are divided into two categories
96 depending on whether they participate in mismatch interactions or not. Since non-mismatched
97 bases are thermodynamically neutral in the Turner model (Turner & Mathews, 2010), their
98 nature should not be a concern, but in practice common puzzle-solving wisdom suggests to make
99 these domains A-rich; for these bases, NEMO uses a 93% A, 1% U, 5% G and 1% C probability

100 distribution. Mismatched bases however do affect the Gibbs free energy contributions of loops
101 and are therefore highly relevant for finding solutions. In such cases, NEMO uses heuristics
102 (described in Fig. 3) derived from [Eterna game playing experience](#).

103

104 **Cost function**

105 The scoring of the samples is a composite function of:

- 106 • the base pair distance (BPD) between the Minimum Free Energy (MFE) structure of the
107 sample sequence as calculated by the folding engine and the target structure, expressed as

$$1 - \frac{(\text{base pair distance})}{2(\text{num target pairs})}$$

- 108 • and the $\Delta\Delta G$ between the MFE of the sample sequence, and its predicted Gibbs free
109 energy in the target conformation, expressed as

$$\frac{1}{1 + (\text{free energy difference})}$$

110

111 **NMCS variants**

112 Two slightly different versions of the NMCS algorithm were implemented: the standard version
113 as found in the GGP literature, and a modified one—that I labeled NMCS-B, standing for
114 “Nested Monte Carlo Search with Best playout policy”—where I introduced an internal
115 maximization mechanism that retains the best scoring playouts throughout the recursion (the
116 difference between them is shown in Fig. 4). Both were executed at the standard level 1 of
117 recursion. Using NMCS recursion at levels 2 or higher would significantly increase
118 computational cost (by up to 30-fold) and was not tested.

119

120 **Selection heuristics for iterations**

121 After the evaluation of a candidate sequence, and provided it was a failure, NEMO identifies the
122 subset of the sequence for which mutations should be considered in preparation for the next
123 iteration by first collecting the indices of all the bases that didn't fold as expected, and then
124 expanding this set with other potentially relevant indices: first adding all mismatch partners of
125 the already collected misfolded positions (as pictured in Fig. 5) and then including closing pairs
126 neighboring pairs that are misfolding by "opening up" (as described in Fig. 6).

127

128 **Testing**

129 In order to test the algorithm and measure its fitness, I repeatedly ran the NEMO tool against the
130 Eterna100 benchmark (Anderson-Lee et al., 2016). Performance and success rates of various
131 builds were measured over 30 single-shot batch runs executed on Stanford University's BioX³
132 and Sherlock clusters. Each process had a default limit of 2500 iterations, corresponding to a
133 maximum of approximately 90 minutes on a single Intel® Core™ i7 3.1 GHz processor for a
134 400 nucleotides long design problem. Separately, MCTS-RNA and NEMO were both tested in
135 the precise conditions used in (Anderson-Lee et al., 2016): up to 5 attempts spanning a maximum
136 of 24 hours.

137

138 **Results**

139 **Self comparisons**

140 Average iteration counts and success rates for the comparison between standard NMCS, NMCS-
141 B and weakened versions of NMCS-B are presented in Fig. 7. The raw data are provided in the
142 Supporting Documents.

143
144 NMCS-B demonstrates a clear superiority over the standard version of NMCS, both by solving
145 about 15 more puzzles on average (92.1/100 to 76.7/100), and by converging to solutions more
146 than twice as fast. As expected, removing algorithmic elements from NEMO causes the
147 performance to worsen: decreasing by about 3 to 6 points in puzzle-solving power, and
148 significantly slowing down convergence by between 15% and 50%. The average success rate of
149 the first NMCS-B pass alone (e.g. solution found with no iterations) is only 43.8/100 though.

150

151 **Comparisons with other engines**

152 When coupled with the standard form of NMCS, the performance of NEMO (77/100) compares
153 to that of the best solving engines previously tested on the same benchmark, the top contender so
154 far being SentRNA with 80/100. But when using the NMCS-B variant, NEMO surpasses them
155 all by a comfortable margin, solving 95 puzzles out of 100 (Fig. 8), and clearly outclasses the
156 other bandit-based method (UCT) implemented in MCTS-RNA which scored 72/100.

157

158 **Discussion**

159 **Unexpected effectiveness**

160 The results point to an overall excellent fitness of the NMCS algorithm when applied to the RNA
161 inverse folding problem. However, the reasons for NEMO's strong performance against the
162 Eterna100 benchmark are not entirely clear.

163

164 The argument that it would be exclusively linked to the quality of the domain knowledge
165 integrated into the tool is at odds with the fact that NEMO only has a limited set of helpful

166 heuristics, lacking for instance special code for dealing with many other known RNA structural
167 complexities like “zigzags” (example depicted in Fig. 2f of (Anderson-Lee et al., 2016)).
168 Furthermore, a direct test by depriving NEMO of heuristics reduces its performance only
169 slightly.

170
171 Also, even though the scoring function in NEMO appears to be novel (Gibbs free energies have
172 been used as selection criterion (Hampson, Sav & Tsang, 2016) but in absolute rather than
173 relative terms) compared with those implemented in other RNA design software packages, its
174 output smoothness over the search spaces remains insufficient to possibly explain any substantial
175 sensitivity improvement in the exploration process, which is supported by the data showing a
176 definite but only modest improvement from using $\Delta\Delta G$.

177
178 Finally, the effectiveness of NMCS versus UCT is indeed known in the General Game Playing
179 field to be game-dependent (Mehat & Cazenave, 2010). Though the poor score of NEMO’s first
180 NMCS phase taken alone in the Eterna100 benchmark test also contradicts the hypothesis that
181 the RNA inverse folding game could simply be an excellent fit for the NMCS algorithm.

182 183 **Imitating human RNA designers**

184 One possible explanation for NEMO's performance is that it partially imitates the solving style of
185 some of the successful human players on the Eterna game platform. In broad terms, the two main
186 classes of puzzle-solving styles delineate a global-local dichotomy perceptible in recorded move
187 histories. Thanks to the fact that the $\Delta\Delta G$ can only be measured when the target structure is
188 entirely filled with valid base pairs, the paths lengths in Fig. S1B allow deriving behavioral

189 information. For instance, the Eterna players who produced the solutions 2, 5, 7 and 9 had filled
190 their canvas early with valid pairs everywhere, while those who generated solution 1, 3 and 4
191 didn't solve the puzzle any faster than the others, but rather favored an incremental method by
192 dividing the problem into smaller ones and by stabilizing first each subdomain before tackling
193 the next one. For that reason, their $\Delta\Delta G$ "tracks" are much shorter than others. Since NEMO
194 uses the $\Delta\Delta G$ in its scoring function, it needs to mimic the "globalist" approach. As for mutating
195 bases or reorienting pairs within or near the misfolded domains, this behavior is common to all
196 Eterna players, and NEMO roughly imitates it in the subset-defining and random-picking phases
197 done in preparation of the next iteration

198
199 As the data show, the NMCS procedure alone is often insufficient to provide an immediate
200 solution to RNA puzzles, which parallels the fact that even expert Eterna players are unlikely to
201 solve a hard puzzle in a single shot. For instance, only 45 players out of the 250,000 registered
202 on Eterna (as of 2018) have solved the ["Snowflake 4" puzzle](#) in the Eterna100 benchmark. The
203 fastest solver on record still required 11 minutes and 345 mutations to complete the challenge,
204 almost 200 more operations than is needed to produce a minimally valid sequence for this
205 puzzle. However, the purpose of the NMCS phase in NEMO is not to solve hard puzzles
206 instantaneously, but to provide reasonable candidate sequences for the overlying explorative
207 process. Qualitatively superior samples tend to benefit any Monte Carlo approach, provided their
208 computational costs stay reasonable and on the condition that the introduced biases leave the
209 relative weights of the visited nodes and subtrees mostly unaffected (James, Konidaris &
210 Rosman, 2017). I encoded into NEMO parts of the domain knowledge (Fig. 3) I acquired by

211 practice and by reading numerous [guides authored by fellow Eterna players](#), and as evidenced by
212 the data presented here, it enhances its NMCS playout policy.

213
214 The decision to combine base pair distances and $\Delta\Delta G$ in the scoring function was both a
215 reflection of my personal puzzle-solving style and the result of analyses on player-submitted
216 solutions and their move histories. Figure S1 conveys that the base pair distance measurement,
217 which is the measurement of choice for the vast majority of RNA inverse folding packages, is a
218 turbulent variable that stays chaotic arbitrarily close to the end goal (an example of which is
219 depicted in Fig. 5). In contrast, no matter the players' puzzle-solving style, global or incremental,
220 the $\Delta\Delta G$ measurement seems much more reliable as an indicator for approaching a solution. The
221 lower performance of the $\Delta\Delta G$ scoring alone (without base pair distance) was predictable: RNA
222 strands can reach multiple conformations, and without a solid structural indicator, an RNA
223 solving engine can waste iterations chasing after a flawed construct that keeps tending to fold
224 into alternate conformations. NEMO currently has no routines to perform local free energy
225 optimizations, but incorporating the $\Delta\Delta G$ in the global scoring function as a cofactor of the base
226 pair distance presumably helps guide the search in the right direction, and the data support its
227 appreciably positive effect.

228
229 During adaptive random walks or tree explorations in RNA puzzle-solving, simple common
230 sense prescribes to keep the parts that fold correctly and only mutate the bases and pairs
231 belonging to domains that do not. RNA inverse folding engines follow this guideline when their
232 main measuring stick is the base pair distance, just as human experts generally do: for instance in
233 the move histories of solutions to the Eterna100 benchmark puzzle titled [“Methaqualone](#)

234 [C₁₆H₁₄N₂O Structural Representation](#)”, expert players mutated misfolded bases 82.8% of the
235 time on average (data provided in Supporting Documents). In that particular phase of the solving
236 process, NEMO’s locality-based heuristics represent a crude but effective approximation of this
237 human behavior.

238
239 However, the current implementation of NEMO lacks a mechanism for imitating a prominent
240 behavior of successful RNA puzzle solvers: backtracking. Data collected on the same previously
241 mentioned Eterna100 puzzle also indicate an average backtracking rate of 22.3%. Presumably,
242 Eterna game players would oftentimes find themselves at a stage of the puzzle-solving process
243 that they regard as “close to solving”. They would then carefully explore various branches of
244 possible mutations, and undo their unsuccessful changes to come back to the previously found
245 satisfactory state if the test was inconclusive. A similar behavior could be implemented in
246 NEMO by replacing the iterated random walk by a judiciously crafted form of tree search.

247
248 **Generality of approach**
249 Concrete applications of the RNA design problem usually require additional constraints like a
250 specified GC content ratio, typically to precisely control melting temperatures for experiments
251 using amplification by polymerase chain reactions (PCR) (Saiki et al., 1988). I intentionally
252 ignored this specific goal in this work so as to better focus on the primary one: solve challenging
253 RNA puzzles. The GC content control goal is trivial to achieve in a post-processing phase.
254 Examples of such algorithms, which explore the neighborhood of a given solution and gradually
255 change its GC/AU/GU pairs ratio, already exist in EternaScripts written by Eterna players
256 ([“Jnicol’s - Remove the GCs v2” by mat747](#)).

257
258 In contrast, designing riboswitches is oftentimes a dissimilar endeavor. Structural constraints
259 usually apply only to small domains within the design space, like binding sites for ligands or
260 oligonucleotides, signaling domains, and gene expression initiation sequences. Structural
261 freedom is granted for the rest of the construct. Should the need arise for a riboswitch with two
262 (or more) precise target conformations, for instance for a nanostructural application, NEMO
263 would have to be modified to properly handle “chain reactions”, i.e. the causal cascade of
264 purines and pyrimidines pairing with different partners over multiple target structures. I already
265 implemented such an algorithm in a previous work, a puzzle-solving bot ([Eterna profile of](#)
266 [ViennaUCT](#)) competing on the Eterna game platform.

267

268 **Conclusion**

269 Until recently, the potential of Monte Carlo techniques applied to the RNA design problem had
270 remained mostly unexplored. The first implementation of the UCT algorithm in this context
271 achieves a good performance against a collection of unyielding RNA puzzles that even human
272 solvers struggle to complete. Though, the Nested Monte Carlo Search algorithm, enhanced by
273 heuristics, outperforms both UCT and all other in silico RNA design approaches. Given the
274 presented encouraging results, the Nested Monte Carlo Search, combined with a novel cost
275 function formula and with a large extent of the accumulated knowledge of Eterna’s expert RNA
276 designers, appears to be a promising technique worthy of deeper investigation by RNA design
277 package creators.

278

279 **Data and materials availability**

280 The source code of NEMO is available at <https://simtk.org/projects/nemo>. Raw tests results and
281 moveset analysis are available at <https://doi.org/10.6084/m9.figshare.6358625>.

282

283 **Acknowledgments**

284 The author would like to thank Rhiju Das, Benjamin Keep and Michelle Wu for their help with
285 the manuscript, and Stanford University for granting him access to the BioX³ and Sherlock
286 clusters.

287

288 **References**

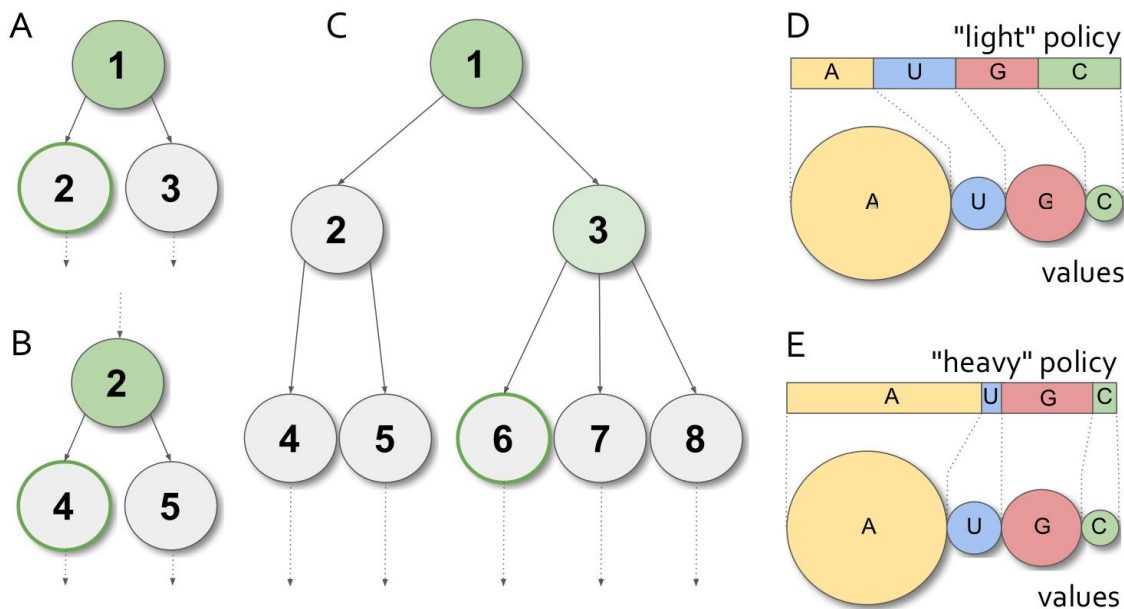
- 289 Anderson-Lee J., Fisker E., Kosaraju V., Wu M., Kong J., Lee J., Lee M., Zada M., Treuille A., Das R.,
290 Eterna Players. 2016. Principles for Predicting RNA Secondary Structure Design Difficulty. *Journal*
291 *of molecular biology* 428:748–757. DOI: 10.1016/j.jmb.2015.11.013.
- 292 Andronescu M., Fejes AP., Hutter F., Hoos HH., Condon A. 2004. A new algorithm for RNA secondary
293 structure design. *Journal of molecular biology* 336:607–624. DOI: 10.1016/j.jmb.2003.12.041.
- 294 Bonnet É., Rzażewski P., Sikora F. 2017. Designing RNA Secondary Structures is Hard.
- 295 Busch A., Backofen R. 2006. INFO-RNA—a fast approach to inverse RNA folding. *Bioinformatics*
296 22:1823–1831. DOI: 10.1093/bioinformatics/btl194.
- 297 Cazenave T. 2009. Nested Monte-Carlo Search. In: *IJCAI*. 456–461.
- 298 Gelly S., Silver D. 2008. Achieving Master Level Play in 9 x 9 Computer Go. In: *AAAI*. 1537–1540.
- 299 Hampson DJD., Sav S., Tsang HH. 2016. Investigation of Multi-Objective Optimization criteria for RNA
300 design. In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8. DOI:
301 10.1109/SSCI.2016.7850232.
- 302 Hofacker IL. 2003. Vienna RNA secondary structure server. *Nucleic acids research* 31:3429–3431. DOI:
303 10.1093/nar/gkg599.
- 304 James S., Konidakis G., Rosman B. 2017. An Analysis of Monte Carlo Tree Search. In: *AAAI*. 3576–
305 3582.
- 306 Kocsis L., Szepesvári C. 2006. Bandit Based Monte-Carlo Planning. In: *Machine Learning: ECML 2006*.
307 Springer Berlin Heidelberg, 282–293. DOI: 10.1007/11871842_29.

- 308 Lee J., Kladwang W., Lee M., Cantu D., Azizyan M., Kim H., Limpaecher A., Yoon S., Treuille A., Das
309 R., EteRNA Participants. 2014. RNA design rules from a massive open laboratory. *Proceedings of*
310 *the National Academy of Sciences of the United States of America* 111:2122–2127. DOI:
311 10.1073/pnas.1313039111.
- 312 Lemieux S., Major F. 2002. RNA canonical and non-canonical base pairing types: a recognition method
313 and complete repertoire. *Nucleic acids research* 30:4250–4263. DOI: 10.1093/nar/gkf540
- 314 Lorenz R., Bernhart SH., Höner Zu Siederdisen C., Tafer H., Flamm C., Stadler PF., Hofacker IL. 2011.
315 ViennaRNA Package 2.0. *Algorithms for molecular biology: AMB* 6:26. DOI: 10.1186/1748-7188-6-
316 26.
- 317 Mehat J., Cazenave T. 2010. Combining UCT and Nested Monte Carlo Search for Single-Player General
318 Game Playing. *IEEE Transactions on Computational Intelligence in AI and Games* 2:271–277. DOI:
319 10.1109/TCIAIG.2010.2088123.
- 320 Ramanujan R., Sabharwal A., Selman B. 2011. On the behavior of UCT in synthetic search spaces. In:
321 *Proc. 21st Int. Conf. Automat. Plan. Sched., Freiburg, Germany*.
- 322 Saiki RK., Gelfand DH., Stoffel S., Scharf SJ., Higuchi R., Horn GT., Mullis KB., Erlich HA. 1988.
323 Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase. *Science*
324 239:487–491. DOI: 10.1126/science.239.4839.487.
- 325 Shi J., Das R., Pande VS. 2018. SentRNA: Improving computational RNA design by incorporating a prior
326 of human design strategies.
- 327 Silver D., Schrittwieser J., Simonyan K., Antonoglou I., Huang A., Guez A., Hubert T., Baker L., Lai M.,
328 Bolton A., Chen Y., Lillicrap T., Hui F., Sifre L., van den Driessche G., Graepel T., Hassabis D.
329 2017. Mastering the game of Go without human knowledge. *Nature* 550:354–359. DOI:
330 10.1038/nature24270.
- 331 Taneda A. 2015. Multi-objective optimization for RNA design with multiple target secondary structures.
332 *BMC bioinformatics* 16:280. DOI: 10.1186/s12859-015-0706-x.
- 333 Turner DH., Mathews DH. 2010. NNDB: the nearest neighbor parameter database for predicting stability
334 of nucleic acid secondary structure. *Nucleic acids research* 38:D280–2. DOI: 10.1093/nar/gkp892.
- 335 Yang X., Yoshizoe K., Taneda A., Tsuda K. 2017. RNA inverse folding using Monte Carlo tree search.
336 *BMC bioinformatics* 18:468. DOI: 10.1186/s12859-017-1882-7.
- 337 Zadeh JN., Steenberg CD., Bois JS., Wolfe BR., Pierce MB., Khan AR., Dirks RM., Pierce NA. 2011.
338 NUPACK: Analysis and design of nucleic acid systems. *Journal of computational chemistry*
339 32:170–173. DOI: 10.1002/jcc.21596.

340

341 **Figures**

342

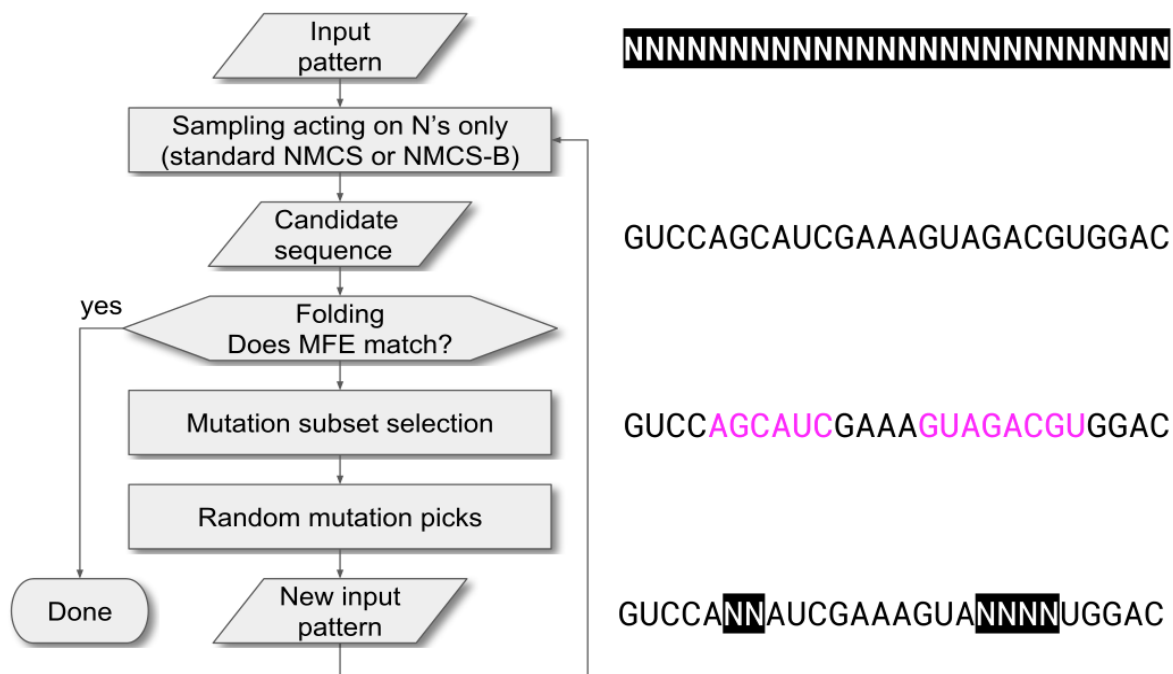


343

344 **Figure 1. NMCS levels and playout policies.** (A) Level-1 NMCS procedure: in an hypothetical
345 game, state 1 has two legal moves, leading to states 2 and 3. The game is played out randomly
346 with both options, the best scoring random game (here with state 2) is selected, and state 2
347 becomes the new root state. (B) The same procedure is applied until the game is finished. (C)
348 The level-2 NMCS procedure is similar and tests all grand-children nodes rather than the direct
349 children ones. States 4 to 8 are all played out randomly. Here, state 3 becomes the root state for
350 the next iteration. (D) Playout policies are applied to choices in the selection of a move in a
351 Monte Carlo playout. Equiprobability is the simplest form of probability distribution. (E) The
352 quality of the sampling can be strongly influenced by “heavy” playouts: using heuristics (which
353 cost CPU resources), the software makes an educated guess as to which option is more valuable.
354 The best playout policies are those that guess the correct order of preference in the available

355 moves. In this example, the “heavy” playout policy makes a decent guess, even though it
356 produces A>G>C>U when A>G>U>C would have been preferable.

357

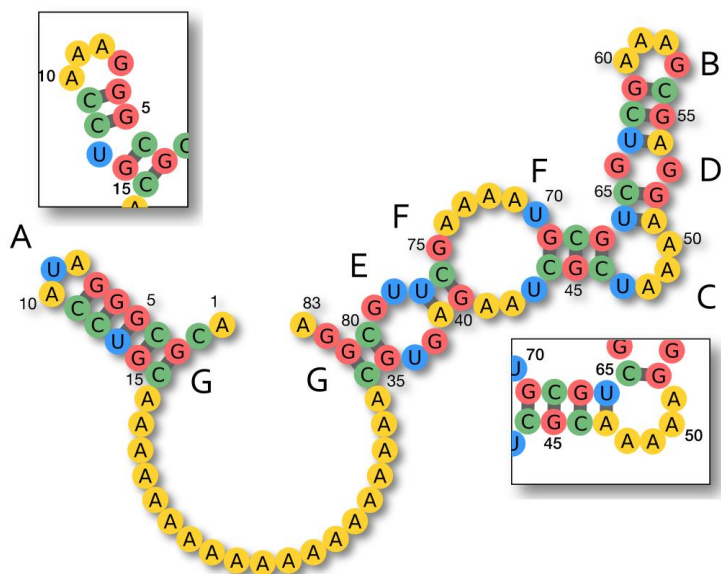


358

359 **Figure 2. Schematic of NEMO's algorithm**, with an example of the evolution of the internal

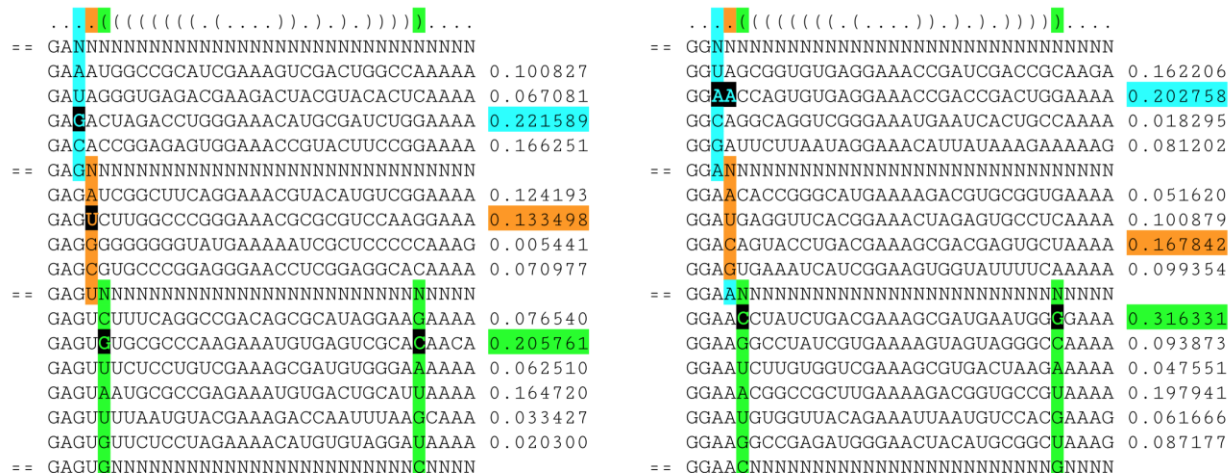
360 state over the first iteration.

361



362
363 **Figure 3. Playout policy heuristics for unpaired mismatched bases.** (A) “blocking” trick
364 occasionally required or simply helpful in some triloops, the inset demonstrates the “sliding” that
365 happens with the U9A mutation (B) standard G/A mismatch in apical loops (C) “blocking”
366 applied to a bulge, the inset demonstrates the unwanted pairing occurring with the U47A
367 mutation (D) standard G/G mismatch in symmetric 1-1 internal loops (E) standard UG/UG
368 combo-mismatch in symmetric 2-2 internal loops (F) some typical favorable mismatches
369 (“boosts”) in internal loops, here A/G and U/U (G) favorable mismatches for external loops and
370 junctions, C/A for GC closing pairs, A/G for CG closing pairs.

371



372

A

B

373 Figure 4. Algorithmic difference between standard NMCS and NMCS-B. Top row, the target

374 structure in dot-bracket notation. In both cases, the algorithm processes positions in order: cyan,

375 orange, green, etc. (A) “vanilla” Nested Monte Carlo Search (NMCS) considers all possible

376 choices at each position by doing one Monte Carlo sampling, and always picks the best of these

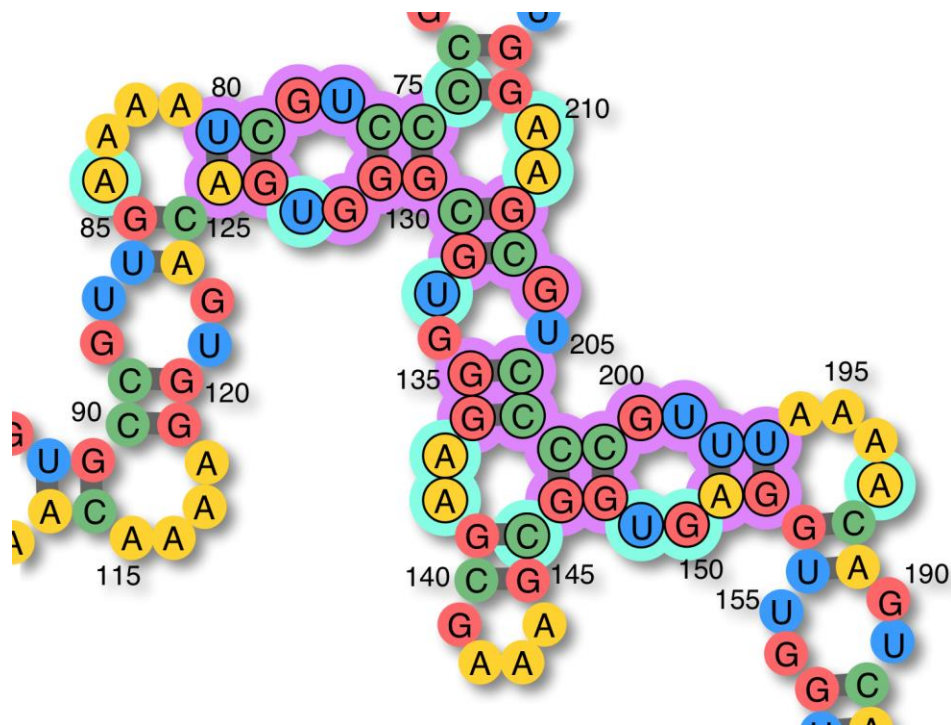
377 outcomes. (B) NMCS-B differs in that it retains the best playout so far and compares it to the

378 samples generated at each recursion step. Here NMCS-B ignores the option C at the orange

379 position, because a better cyan sample playout is known. At the green step, the sample playout

380 sporting the CG pair becomes the best sample known so far and might influence the next steps.

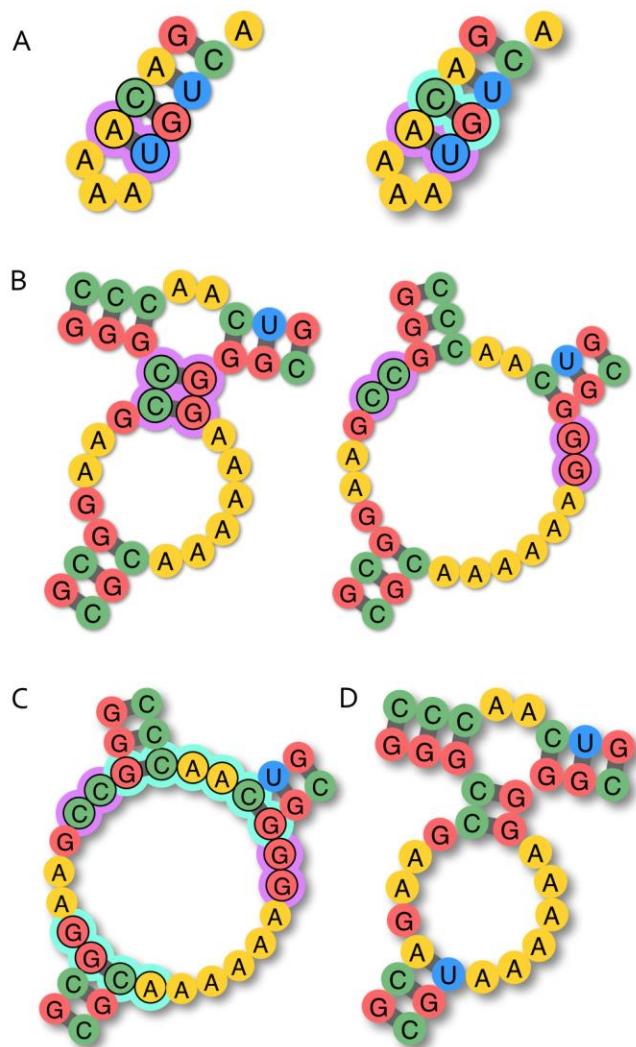
381



382

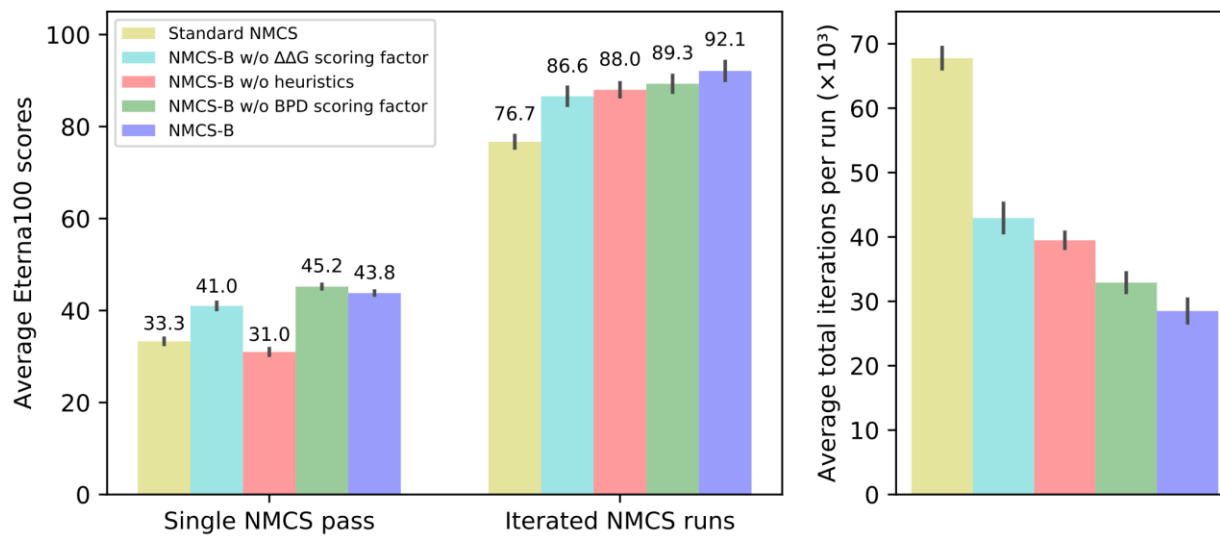
383 **Figure 5. Mutation candidates subset selection.** This program phase begins with including all
384 misfolded bases (purple marks). Additionally, their mismatch partners (cyan marks) are included
385 as well. Note: in this example (which has a base pair distance of 24 w.r.t. the target structure),
386 the A84G and A210C mutations both stabilize the puzzle completely. In other words two
387 solutions exist only 1 one-point mutation away from this particular sequence.

388



389
390 **Figure 6. Heuristic rule of including closing pairs around pairs misfolding by “opening**
391 **up”.** (A) An AU pair closing a triloop cannot hold if the enclosing pair is GC/CG. Mutating the
392 closing pair to CG/GC would likely solve the issue, but at this stage, the goal is to collect all
393 mutations that could. Changing the enclosing pair to UA could solve the local problem as well,
394 therefore the cyan marked pair should be included in the subset. (B) A slightly more complex
395 case with a short stem linking two loops. (C) The pairs closing the large junction in the
396 misfolded structure, and their associated mismatches (cyan marks) should all be added to the
397 mutation candidates list. (D) In this particular case, it turns out that no mutations of the purple-

398 marked pairs and no alternate boosting of the surrounding loops can help the short helix to hold
399 in place. The only mutation that works here, is to change the bottom-left closing pair to AU/UA.
400

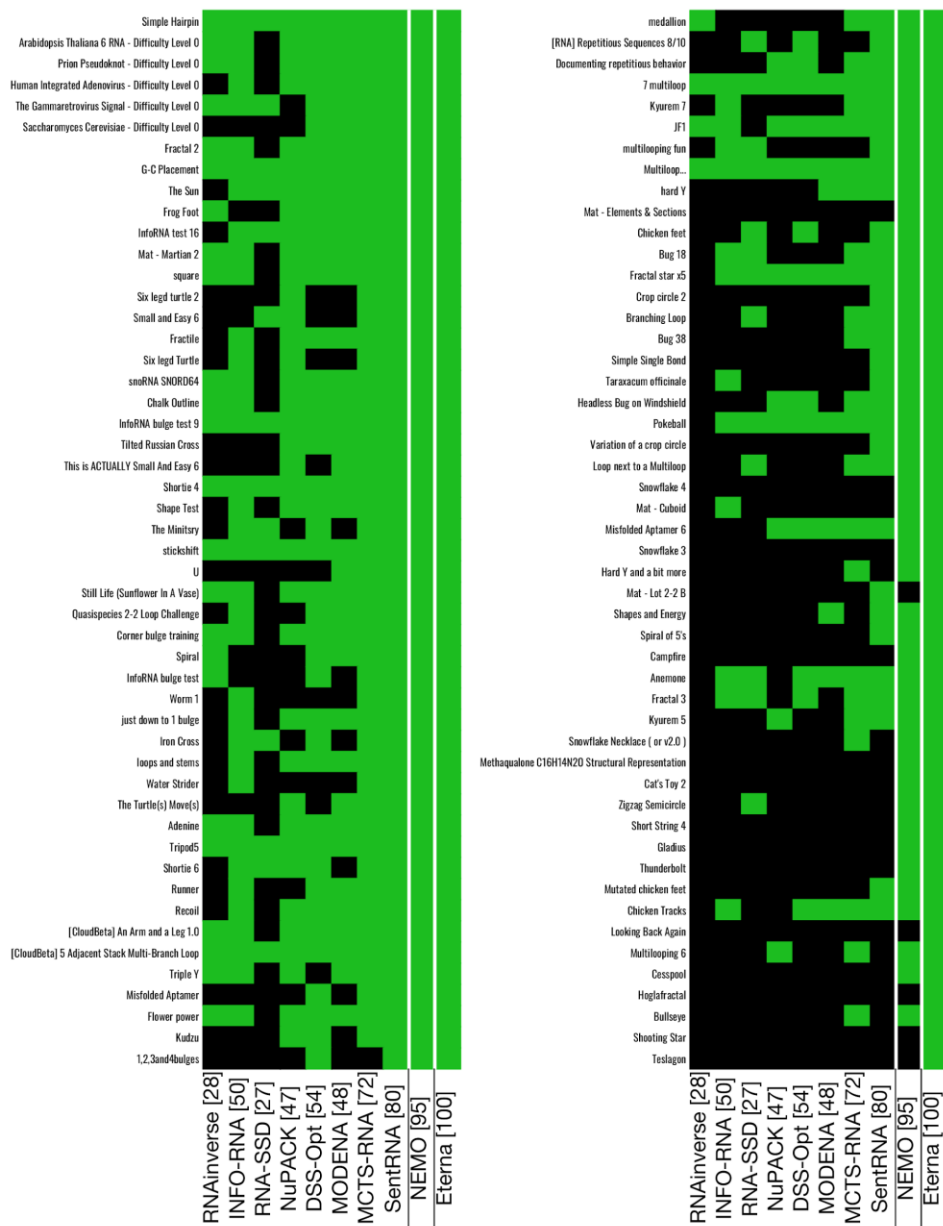


401

402 **Figure 7. NEMO performance tests.** Average scores and iteration counts over 30 single

403 attempts runs against the Eterna100 benchmark.

404



405

406 **Figure 8. Comparative Eterna100 benchmark results.** Black squares are failures, green ones

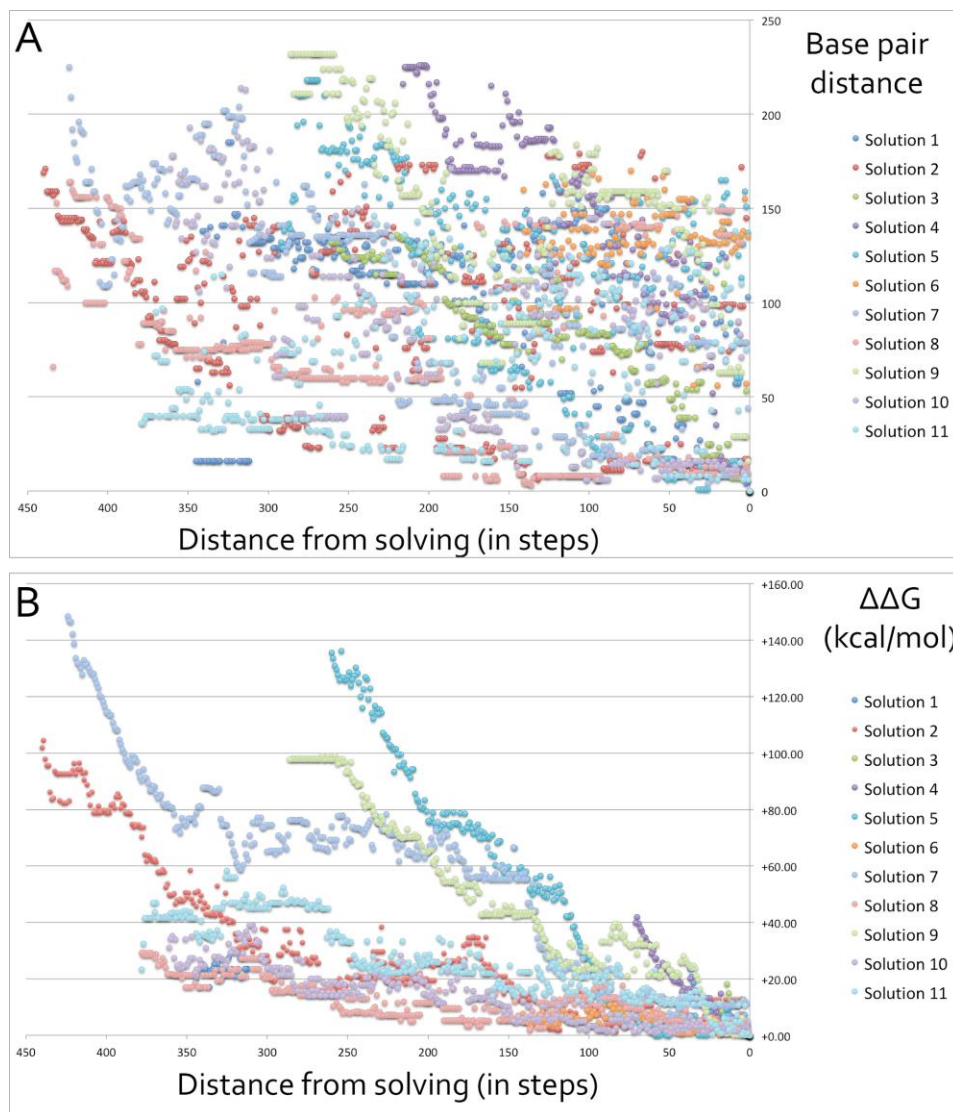
407 are successes. RNAinverse, INFO-RNA, RNA-SSD, NuPACK, DSS-Opt, MODENA et Eterna

408 players results data were taken from Anderson-Lee et al. 2016, SentRNA data taken from Shi et

409 al. 2018. MCTS-RNA was configured to ignore GC content requirements. NEMO was run with

410 NMCS-B active.

411 **Supplementary Figures**



412
413 **Figure S1. Base pair distances and free energy differences as functions of distance in**
414 **mutation steps in human solving.** Both graphs relate to solutions provided by human experts
415 for the [“Snowflake 4” puzzle](#) of the Eterna100 benchmark. Evolution over time (in mutation
416 steps) of (A) base pair distances, and of (B) Gibbs free energy differences ($\Delta\Delta G$) until a solution
417 is found.
418