Poetic RNA: Adapting RNA Design Methods to the Analysis of Poetry

Veronica Dahl and M. Dolores Jiménez-López and Olivier Perriquet

Abstract The style in which an RNA molecule folds in space obeys laws of nucleotide binding and attraction which are encoded in its primary structure, that is, in the sequence of nucleotides conforming it. Natural language sentences can also be viewed as encodings for a structure in space- in this case, a parse tree- which exhibits relationships or bindings between different parts of the sentence. We explore the possibilities in adapting a recent, simple and elegant methodology for bioinformatics which has been successfully used for RNA design, to the problem of parsing poems that follow specific stylistic trends. The methodology introduced in this paper can be easily express in terms of multi-agent systems.

1 Introduction

The application to molecular biology of AI methods such as logic programming and constraint reasoning constitutes a fascinating interdisciplinary field which, despite being relatively new, has already proved quite fertile. For instance, logic programming techniques have been used to describe and analyze protein structure [17], for protein secondary structure prediction [16], for drug design [13] and for predicting gene functions [12].

Veronica Dahl
Departament of Computing Science, Simon Fraser University, Burnaby, B.C., Canada
Research Group on Mathematical Linguistics, Rovira i Virgili University, 43002 Tarragona, Spain,
e-mail: veronica@cs.sfu.ca

M. Dolores Jiménez-López
Research Group on Mathematical Linguistics, Rovira i Virgili University, 43002 Tarragona, Spain,
e-mail: mariadolores.jimenez@urv.cat

Olivier Perriquet
Centre for Artificial Intelligence, New University of Lisbon, Lisbon, Portugal,
e-mail: olivier@perriquet.net

As well, methodologies that pertain to the natural language processing field of AI are now being exploited to analyze biological sequences, which is uncovering similarities between the languages of molecular biology and human languages (e.g. [5]).

On the other hand, the potential influence of computational molecular biology over natural language processing, has been much less studied, although some similarities and cross-fertilization potential are starting to be identified and exploited (e.g. [8]).

In [4], a novel methodology was presented -chrRNA- for addressing an important bioinformatics problem which has been proved to be computationally hard: that of finding an RNA sequence which folds into a given structure.

Obvious potential applications are to in vitro genetics, by enabling the scientists to produce RNAs artificially from sequences; and to drug design, which typically progresses backwards from proteins to RNAs and finally to DNAs.

A much less evident potential application is proposed in this paper: the adaptation of the same methodology to computational linguistics problems. We explore in particular its uses for solving the apparently disparate problem of using stylistic information as an aid for processing poetry, e.g. for parsing or for authorship determination.

Our work springs from the observation that, in the same way as molecules each have their own style, while all obeying the same nucleic acid grammar (that of RNA or DNA), human artistic creations are marked by the specific style of their author even when the grammar available to all authors (the elements they may use, the ways in which they may combine those elements) may be the same. Thus the English language, for instance, obeys the same grammar rules no matter who uses it, but nevertheless, each poet uses that same grammar with very different, while characteristic for each author, styles.

The method developed in [4] for determining what RNA sequence a given molecule's structure folds into involves a very simple grammar augmented by probabilities. These probabilities encode the molecule's "style", as it were, so adapting our method to computational linguistics involves resorting to stylistic probabilities observed in a given author's poetic production in order to aid in the parse of a given poem of the same author, or to aid in determining authorship itself.

2 Background

2.1 Constraint Handling Rules (CHR) and Constraint Handling Rule Grammars (CHRG)

Constraint Handling Rules (CHR) provide a simple bottom-up framework which has been proved to be useful for algorithms dealing with constraints [10, 7]. Because logic terms are used, grammars can be described in human-like terms and are

powerfully extended through (hidden) logical inference. The format of CHR rules is:

```
Head ==>Guard|Body
```

`Head` and `Body` are conjunctions of atoms and `Guard` is a test constructed from (Prolog) built-in or system-defined predicates. The variables in `Guard` and `Body` occur also in `Head`. If the `Guard` is the constant "true", then it is omitted together with the vertical bar. Its logical meaning is the formula $(Guard \rightarrow (Head \rightarrow Body))$ and the meaning of a program is given by conjunction. There are three types of CHR rules:

- *Propagation rules*, which add new constraints (body) to the constraint set.
- *Simplification rules*, which also add as new constraints those in the body, but remove as well the ones in the head of the rule.
- *Simpagation rules*, which combine propagation and simplification traits, and allow us to select which of the constraints mentioned in the head of the rule should remain and which should be removed from the constraint set.

The rewrite symbols for the first two rules are respectively: `==>`, `<=>` and for simgation rules, the notation is `Head1\Head2<=>body`. Anything in `Head1` remains in the constraint set and anything in `Head2` is removed from the constraint set.

CHR have a grammatical counterpart -CHRG [7]- which is to CHR what Definite Clause Grammars are to Prolog: grammar symbols compile into constraints in which the word boundaries are made explicit automatically, so there's no need to handle them in the grammar. The notation that distinguishes a grammar rule from a plain CHR rule. is `::>`

Thus for instance, the following CHRG rule:

```
verb, noun_phrase==> verb_phrase.
```

compiles into a CHR counterpart in which the start and end points are visible:

```
verb(Start,P1),
noun_phrase(P1,End)==> verb_phrase(Start,End).
```

2.2 The chrRNA Method for RNA Design

Research on the language of nucleic acid, parsimoniously but powerfully formed basically with only four letters, or nucleotides (A, C, T and G), has in the past few years started to uncover fruitful similarities between this language and human languages [19]. Just as a sentence in English is only the visible part, or tip of the iceberg, for that sentence's rich structure, which is essential to its meaning, a nucleic acid "sentence" (i.e., a sequence of A's, C's, T's and G's) hides, or codes for, an incredibly rich structure which takes 3D shape when some of the nucleotides in the sentence bind with others, and which is just as essential to that sentence's meaning.

Scientists in computational molecular biology are interested both in the generation aspect of this process, namely predicting which 3D structure a known sequence of nucleotides might fold into, and in its analyzing aspect, namely finding out what sequence of nucleotides codes for a molecule whose 3D structure is known.

Both these aspects have approximate computational solutions, as long as compromises are made with a particularly hard subproblem: that of molecules containing a specific 3D structure called a *pseudoknot*. Simple pseudoknots occur when a structure of the form of a hairpin loop contains nucleotides that bind outside the loop.

Methods dealing with pseudoknots are delicate to conceive. Initially, the methods for RNA structure prediction used to simply disregard the possibility of having pseudoknots, due to the huge algorithmic complexity these would bring to the problem, compared to the moderate gain in prediction accuracy. For the analysis aspect, also known as the *inverse RNA problem*, or the problem of *RNA design*, the same algorithmic issue with pseudoknots persists. The solution proposed recently in [4] allows pseudoknots.

Previous solutions to this problem divide the whole structure into smaller substructures and apply some techniques to resolve it for smaller parts, which causes them to be slow while working with longer RNAs (more than 500 bases). Bavarian and Dahl show in [4] that by using a set of simple Constraint Handling Rules, this problem could actually have an approximate but somehow useful solution in linear time, despite the simplistic grammar model.

This solution- named chrRNA- encodes an RNA molecule's style in terms of probabilities which are incorporated into the (otherwise highly ambiguous) grammar rules that describe RNA primary structure. With the aid of their embedded probabilities, they guide the assignment of content (one of A, C, U[1] or G) to nucleotides we know are paired but whose identity is not known, and also for the nucleotides we know are unpaired but whose identity is likewise unknown.

These rules are based on the simple but highly ambiguous context free grammar proposed in [1] for RNA secondary structure prediction:

$$S \rightarrow cSg|gSc|aSu|uSa|gSu|uSg$$

$$S \rightarrow aS|gS|uS|cS$$

$$S \rightarrow a|g|u|c$$

$$S \rightarrow SS$$

Using CHR to implement this grammar allows us both to exploit the bottom-up characteristic of CHR rules, as well as keep track of ambiguous readings with no special overhead.

[1] In RNA, thymine (T) is replaced by uracile (U), but U and T are perfectly equivalent at the informational level.

2.2.1 Representation issues

As mentioned, the problem consists of finding a sequence of A, C, U and G which folds into the (given as input) structure of the desired RNA. In the chrRNA method, this structure is entered in the format of CHR constraints, e.g. for expressing that the base number 1 and the base number 43 in the sequence are paired together, we add the constraint `pair(1,43)` or if base number 3 is unpaired, the corresponding constraint would be `upair(3)`. One advantage of this input format to the input format used by previous methods (RNAinverse and RNA-SSD) is that it is capable of accepting pseudoknots in the input structure.

2.2.2 Processing issues

We now need to assign nucleotides to each position given the input constraints. The trivial solution of randomly assigning one of the Watson-Crick pairs (an AU or CG pair) to each base pair and one of the four nucleotides (A, C, G and U) to the unpaired bases is not a reliable one, since we might end up with a sequence that may not actually fold into the input structure. This is because the number of GC pairs has an important role in stabilizing a certain structure. For instance, if we assign base *G* to 1 and base *U* to position 43 and if we have a base *C* in position 42, in the end, the structure might be `pair(1,42)` instead of `pair(1,43)`.

The solution we have proposed uses CHR rules combined with the probabilities that are believed to govern the proportion of base pairs within RNA sequences, calculated by comparing several RNAs together from Gutell lab's comparative RNA website, a database of known RNA secondary structures. After comparing 100 test cases with various length from 100 to 1500 bases, they found the following probabilities for each base pair:

$$P_{CG} = 0.53, P_{AU} = 0.35, P_{GU} = 0.12$$

The other probabilities which are of interest are the probabilities for an unpaired base to be one of *A*, *C*, *G*, or *U*. The results are as follows:

$$P_G = 0.18, P_A = 0.34, P_C = 0.27, P_U = 0.20$$

Inserting the probabilities into the rules is done by generating a random variable in the guard section of the rules, and then testing this random variable according to the probabilities: for instance if the random variable `I` generated when trying to instantiate a pair of nucleotides X1 and Y21 is less than 0.53, the grammar rule will assign a GC pair to `pair(X1,Y1)`.

The result is a quite simple while powerful system which performs better than the two previous methods for longer sequences, and which is capable of handling pseudoknots. The application of our ideas could be useful for in vitro genetics and for drug design. We shall next argue that they can also be fruitfully adapted for

literary style processing, and exemplify this thesis around the poetry of the Cuban author Nicolas Guillen.

3 Poetic Style

Just as the CF grammar of RNA by itself is highly ambiguous, human languages in general and poems in particular are also prone to ambiguity; however we have chosen poems as a starting point because poetic style often departs in idiosyncratic ways from standard word orderings often found in prose, and identifying such style trends might therefore contribute clues that serve to disambiguate, much in the way as the probabilities used in chrRNA help identify an RNA string unambiguously. We can moreover express these trends precisely in the same way: through probabilities resulting from an analysis of the particular author's style.

As an example, Guillen's poems make frequent use of topicalization, which often makes it hard to identify a sentence's subject from syntax alone. For instance, whereas in Spanish subjects precede the verb, a verse could start with the main verb, *followed* by the subject, as in: "Dejó el borracho en su coche, dejó el cabaret..." which reordered and translated, would be: "The drunkard left the cabaret ...in his car". In other cases, of course, what follows the main verb is the more expected direct object, as in "dejó el cabaret".

However, an analysis of Guillen's style might allow us to encode into the context free rules, just as we did for nucleic acid rules, probabilities that can help us disambiguate, e.g. by helping us determine in which cases an NP following a verb is its subject (as "el borracho") and in which cases it is an object (as "el cabaret"). We might want to encode Spanish and stylistic observations such as:

- a verb's subject has high probability of being adjacent to (either preceding or following it) the main verb;
- a direct object has high probability of following, rather than preceding, the verb;
- a main verb which is repeated has high probability of having in its repeated occurrence the same, albeit implicit, subject as the first occurrence's;
- a second verb with no overt subject, particularly if in the same tense and person as the first verb, has high probability of implicitly having the same subject as the first verb.

According to these rules, we can bet that in our example, "dejó el cabaret", which has no overt subject, actually means "the drunkard left the cabaret", rather than "the cabaret left".

Of course, the more accurate an analysis of a given poet's style and of the exact values of the different probabilities - which above are only stated as being "high" -, the more accurate the proposed method will be for correctly using style to interpret a poem's meaning. It is not our purpose in this paper to propose precise values for any such probabilities; rather, we aim at showing how the chrRNA method that has proved so successful for RNA design can be adapted also to literary analysis.

4 PoeticRNA

We now abstract the problem of analyzing poems to make it amenable to the same form as that of RNA design.

4.1 Representation issues

Our "nucleotides" are now blocks of words marked by syntactic functions, such as verb, noun phrase, preposition phrase, which can in a first stage be gleaned through a regular CHRG , e.g. "el borracho" can be analyzed into a noun phrase stretching from position 2 in the input sentence to position 3 but whose role inside the sentence- i.e., the style which binds concepts with one another- remains to be found. This shows up in a CHR rule as the constraint `noun_phrase(2,3,[el,borracho])`, and in its CHRG counterpart, as the grammar symbol `noun_phrase([el,borracho])`, with word boundaries left implicit (but accessible if needed).

The roles we are after (subject, direct object, etc.) will be represented also through grammar symbols that compile into constraints. The roles will be superimposed, so to speak, over the stretch of input string concerned. Thus if we view "noun phrase" as a label covering the substring "el borracho", once we find that this noun phrase's role is that of subject, we will add a new constraint which, in this same graphic view, would label the same substring with "subject".

Other arguments relevant to the analysis can of course be included, for our purposes here we'll only add in some cases an index I which will be an identifier to the "nucleotide" that the block containing it attaches to, or refers to (e.g. to the antecedent in the case of a a prepositional phrase or a relative clause).

4.2 Processing issues

Let us consider the following sentence, adapted from Nicolas Guillen's "La Guitarra" for explanatory purposes: "Dejó el borracho en su coche, dejó el cabaret sombrio". This can be parsed into one sentence, which explicitly and in English would correspond to "The drunkard left the sombre cabaret (by traveling) in his car". Any Spanish reader would understand that the verb's repetition is for poetic effect, rather than a "new" main verb. A machine analyzer, however, would recognize two sentences, corresponding either to: "(Someone) left the drunkard in his car, and (the same person) left the sombre cabaret, or "Someone left the drunkard in his car, and the sombre cabaret left". The first interpretation is likely when one considers that implicit subjects are very common in Spanish (so any reasonable Spanish analyzer would conceive it); the second one is nonsensical for humans but plausible from syntax alone, and therefore, a fair candidate for a poetically uninformed parser. Note

that while the state-of-the-art in parsing would allow us to choose between these two interpretations, perhaps by paying the price of including semantic type information to preclude non animated subjects such as “the cabaret” for movement verbs such as “dejó”, the interpretation as a single sentence with verb repetition would remain inaccessible, to the best of our knowledge, to state-of-the-art parsers, including those meant to analyze poetry.

As in the case of RNA design, we can encode probability values that will allow us to determine, in case of ambiguity, which possible analysis is more likely. Thus if we’ve encoded probabilities to the effect that when a verb is not the initial word in a sentence, the noun phrase that follows it is likely to be a direct object rather than a subject, the first rule below will capture that noun phrase into the verb phrase (by creating a `verb_phrase` symbol that spans both substrings, i.e. from P1 to P3), while marking it as the direct object of that verb (by associating the same index, J, to both the verb and the object). This rule will be used for instance for “dejó el cabaret sombrío”. Notice that we express it as a CHR rule in order to access the word boundaries and generate the new symbols with appropriate span.

If our probabilities moreover indicate that initial verbs in Guillen’s poetry are likely to appear before the subject noun phrase just for stylistic effect, the second of the following two rules will be taken for “dejó el borracho”. This rule records the role of subject found for this noun phrase and marks both it and the verb with an index I, to indicate that the string L2 is the subject of the verb L1. Notice that since we don’t need word boundaries made explicit, this rule is a CHRG one- the word boundaries will be added automatically at compilation time.

```
verb(P1,P2,L1),
noun_phrase(P2,P3,L2)==>prob(subj_verb_inversion,low),
append(L1,L2,L)  | verb(p1,p2,L1,J),
direct_object(P2,P3,L2,J), verb_phrase(P1,P3,L).
verb(L1),
noun_phrase(L2) ::> prob(subj_verb_inversion,high) |
verb(L1,I), subject(L2,I).
```

Likewise, we could use probabilities to express the likelihood that a repeated verb’s implicit subject refers to the subject of that same verb’s other occurrence, and consult them in the guard of the concerned rule in order to co-index both occurrences of that verb with the same subject.

4.3 Our Methodology as a Multi-Agent System

Multi-agent systems are especially adequate for the solution of problems of dynamic, uncertain and distributed nature. The problem of parsing poems that follow specific stylistic trends –in order to, for example, determine the authorship– is a problem that can be seen as dynamic, uncertain and distributed. Taking into account

this idea, and considering the features of our methodology, we think that it is possible to express the model introduced here in terms of multi-agent systems. Our multi-agent system is basically composed by two agents:

- the *basic parser* whose task is gleaning the basic syntactic structures;
- and the *probabilistic part of the parser* whose task consists on labeling the ambiguous structures with their appropriate role and co-indexing them with the other structures they relate to.

These two agents collaborate in order to deal with the complex task of parsing poems.

5 Concluding Remarks

Algorithmic approaches to poetry have been around for a relatively long time. They mostly focus on generating poetry by automated or semi-automated means (e.g. [15]), and as such, belong to the general field of Electronic Writing. Automated poetry analysis, on the other hand, remains a bit more elusive, and concentrates on the more mechanizable subtasks, such as automated analysis of sound and meter (e.g. [14]).

To the best of our knowledge, this work is the first in approaching poetic analysis through stylistic probabilities added to a constraint-based parser. The exploitable similarities between molecular and poetic style are potentially much bigger than explored in this first paper on the subject, and relatively straightforward to implement in the underlying model we propose. What we just exemplified with a process called topicalization, would gain in being pursued on a corpus of known poetic techniques that make use of parsing ambiguity. In the scope of this paper, note that the ground level of our analysis is syntactic, in that respect we do not consider for instance what may be termed recombinant words or concepts (see [20] for instance) that would be visible only at an infra-syntaxic level. The neologism *simpagation* that appeared in a previous section is such an example of re-invention of language out of the scope of the present analysis, based on an invention process called in french *mots-valises* which are built by blending together two different words. These remarks give a picture of the horizon we assign to our work.

Searls exposes in [19] the important role of computational linguistics techniques in the domain of nucleic acid string analysis. Based on the indubitable efficiency of these techniques in bioinformatics, he terms the processes at work in the production of protein the language of the gene and the language of proteins . The metaphor made its own way and the connotative field of linguistics is currently well anchored in the domain: anyone nowadays speaks about gene and protein *expression*, open *reading* frame, *transcription* factors, the *alphabet* of DNA, gene *translation*, *messenger* RNA, etc. Alternatively the metaphor of film developing, for instance, could have been used with a similar expressiveness (DNA footage being developped into

protein image), but the lack of formalism would hardly have turned it into a operational paradigm: the linguistic denomination is more than a mere metaphor due to the concrete operability of linguistics tools. When speaking about a molecule's style, we actually go one step further in that anthropomorphic projection, and we do it on purpose, in a very prospective manner. Our intention is to to explore a backward genre contamination where the linguistic approach itself gets inspired by its own application to nucleic acid analysis.

References

1. Baldi, P.: Bioinformatics: the machine learning approach. MIT Press, Cambridge (1998).
2. Barranco-Mendoza, A.: Stochastic and Heuristic Modelling for Analysis of the Growth of Pre-Invasive Lesions and for a Multidisciplinary Approach to Early Cancer Diagnosis. Ph.D. Thesis, Simon Fraser University, Burnaby, BC (2005).
3. Barranco-Mendoza, A., Persaoud, D.R., Dahl, V.: A property-based model for lung cancer diagnosis. RECOMB poster, 27–31 (2004).
4. Bavarian, M., Dahl, V.: Constraint-Based Methods for Biological Sequence Analysis. Journal of Universal Computing Science **12(11)**, 1500–1520 (2006).
5. Bel-Enguix, G., Dahl, V., Jiménez-López, M.D.: DNA and Natural Languages: Text Mining. In: Fred, A. Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 140-145. INSTICC, Madeira (2009).
6. Cai, L., Malmberg, R.L., Wu, Y.: Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics **19/1**, i66-i73 (2003).
7. Christiansen, H.: CHR as Grammar Formalism, a First Report. In: Apt, K.R., Bartak, R., Monfroy, E., Rossi, F., (eds.) Sixth Annual Workshop of the ERCIM Working Group on Constraints. Prague (2001).
8. Dahl, V. Maharshak, E.: DNA replication as a Model for Computational Linguistics. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) Methods and Models in Artificial and Natural Computation, pp. 346-355. Springer, Berlin (2009).
9. Eddy, S.R., Durbin, R.: RNA sequence analysis using covariance models. NAR **22/1**, 2079-2088 (1994).
10. Fruhwirth, T.: Theory and Practice of Constraint Handling Rules. Journal of Logic Programming **37(1-3)**, 95-138 (1998).
11. Guillen, Nicolas: Motivos de son (1930). Biblioteca Virtual Miguel de Cervantes, Alicante (2001)
12. King, R.D.: Applying inductive logic programming to predicting gene function. AI Mag. **25-1**, 57–68 (2004)
13. King, R. D., Muggleton, S., Lewis, R.A., Sternberg, M. J. E.: Drug design by machine learning. Proc. Natl. Acad. Sci. **89**, 11322–11326 (1992).
14. Logan, Harry M: Computer Analysis of Sound and Meter in Poetry. College Literature **15/1**, 19-24 (1988).
15. Mendelowitz, E.: Drafting poems: inverted potentialities. In: International Multimedia Conference, Proceedings of the 14th Annual ACM International Conference on Multimedia, pp. 1047-1048. Santa Barbara (2006).
16. Muggleton, S.,King,R.D., Sternberg.,M.J.E.: Protein secondary structure prediction using logic-based machine learning. Protein Eng. **5**, 647–657 (1992)
17. Rawling, C.J., Taylor. W.R., Nyakairo, J., Fox, J., Sternberg, M.J.E.: Reasoning about protein topology using the logic programming language PROLOG. J. Mol. Bio. **3-4**, 151–157 (1985).
18. Rivas, E., Eddy, S. R.: The language of RNA: A formal grammar that includes pseudoknots. Bioinformatics **16**, 334-340 (2000).

19. Searls, D.B.: The computational linguistics of biological sequences. Artificial intelligence and molecular biology, American Association for Artificial Intelligence, 47-120 (1993).
20. Seaman, W.: Recombinant Poetics: Emergent Meaning as Examined and Explored Within a Specific Generative Virtual Environment. Ph.D. Thesis, CAiiA (Centre for Advanced Inquiry in the Interactive Arts), (1999).
21. Woods, W. A.: An experimental parsing system for transition network grammars.In Rustin, R. (ed.) Natural Language Processing, pp. 145-149. Algorithmics Press, New York (1973).