Κεφάλαιο 10: Υπολογιστικές Γραμματικές

Σύνοψη

Στο κεφάλαιο αυτό θα μελετήσουμε μια γενικότερη κατηγορία μεθόδων, τις υπολογιστικές γραμματικές, οι οποίες περιλαμβάνουν σαν ειδικές περιπτώσεις μοντέλα που είδαμε σε προηγούμενα κεφάλαια (πρότυπα, HMM), αλλά και πιο σύνθετες δομές οι οποίες μπορούν να μοντελοποιήσουν καλύτερα μια σειρά από βιολογικά προβλήματα. Θα δούμε τις ιστορικές καταβολές αυτών των μεθόδων και θα μελετήσουμε τις κατηγορίες εκείνες που μπορούν να χρησιμοποιηθούν στην πρόγνωση της δευτεροταγούς δομής του RNA καλύπτοντας και το ζευγάρωμα των βάσεων. Θα μιλήσουμε για τις γνωστές εφαρμογές και το λογισμικό που βασίζεται σε τέτοιες μεθοδολογίες, και στο τέλος θα δούμε και κάποιες εφαρμογές στην πρόγνωση δευτεροταγούς δομής πρωτεϊνών, εφαρμογές που κάνουν ένα πρώτο βήμα στην πρόγνωση των μακρινών αλληλεπιδράσεων.

Προαπαιτούμενη γνώση

Στο κεφάλαιο αυτό απαραίτητη είναι η γνώση των εννοιών των κεφαλαίων που ασχολούνται με τα πρότυπα (κεφάλαιο 4), τα HMM (κεφάλαιο 8), αλλά και τις μεθόδους πρόγνωσης (κεφάλαιο 7).

10. Εισαγωγή

Στο κεφάλαιο αυτό, αφού έχουμε δει λεπτομερώς τις κανονικές εκφράσεις (regular expressions), τα πρότυπα (patterns) στις αλληλουχίες, τα προφίλ (profiles) αλλά και τα Hidden Markov Models (HMMs), θα προχωρήσουμε ένα βήμα παραπέρα. Θα δούμε πώς εντάσσονται τα παραπάνω συστήματα στη μεγάλη κατηγορία των τυπικών γλωσσών που χρησιμοποιούνται στην υπολογιστική γλωσσολογία. Στη συνέχεια, αφού έχουμε μελετήσει τις περιπτώσεις στις οποίες τα απλά αυτά μοντέλα δεν επαρκούν, θα δούμε και παραδείγματα εφαρμογών πιο σύνθετων μοντέλων.

Η θεωρία των τυπικών γλωσσών (formal language theory), ορίζει μια «γλώσσα» ως ένα σύνολο συμβόλων από κάποιο αλφάβητο. Η γραμματική, είναι μια προσέγγιση για τον ορισμό της γλώσσας, η οποία βασίζεται σε ένα σύνολο από κανόνες. Οι κανόνες αυτοί (rewriting rules) παίρνουν τη μορφή όπως $A \rightarrow xB$, όπου τα κεφαλαία γράμματα συμβολίζουν τα προσωρινά, αφηρημένα, μη-τερματικά σύμβολα (nonterminal symbols), τα οποία δεν εμφανίζονται στο αλφάβητο, ενώ τα πεζά γράμματα συμβολίζουν τα παρατηρήσιμα, τερματικά (terminal symbols) σύμβολα, τα οποία υπάρχουν στο αλφάβητο. Ο παραπάνω κανόνας καθορίζει ότι κάθε εμφάνιση του μη-τερματικού συμβόλου A μπορεί να αντικατασταθεί από το τερματικό x και το μη-τερματικό B. Γενικά, ξεκινώντας από ένα μη-τερματικό σύμβολο S, η παραγωγή της γραμματικής συνίσταται σε μια σειρά από τέτοια βήματα αντικατάστασης, τα οποία τερματίζονται όταν το τελευταίο μη-τερματικό σύμβολο έχει εξαφανιστεί.

Η ταξινόμηση και η υπολογιστική μελέτη των τυπικών γραμματικών, οφείλει την ύπαρξή της στον μεγάλο γλωσσολόγο του MIT και διάσημο αναρχικό φιλόσοφο και στοχαστή, Noam Chomsky. Η συνεισφορά του αυτή ήταν ορόσημο για την υπολογιστική γλωσσολογία και οι μεθοδολογίες αυτές χρησιμοποιούνται μέχρι σήμερα, τόσο στη μελέτη των φυσικών γλωσσών, όσο και στη θεωρητική πληροφορική (γλώσσες προγραμματισμού κλπ), αλλά και στη Βιοπληροφορική, όπως θα δούμε στο κεφάλαιο αυτό. Τα περισσότερα περιεχόμενα αυτού του κεφαλαίου, ακολουθούν την ορολογία και τη δομή του αντίστοιχου κεφαλαίου των (Durbin, Eddy, Krogh, & Mithison, 1998) ενώ αναφορές γίνονται και σε αντίστοιχα άρθρα ανασκόπησης, π.χ. (Searls, 2002).

10.1. Η ιεραρχία των γραμματικών του Chomsky

Το 1956 ο Noam Chomsky ταξινόμησε τις τυπικές γραμματικές σε ιεραρχία με κριτήριο τους τύπους των κανόνων παραγωγής τους (Chomsky, 1956). Σύμφωνα με αυτήν την ταξινόμηση μια τυπική γλώσσα G αποτελείται από:

- Ένα πεπερασμένο σύνολο V από μη τερματικά σύμβολα
- Ένα πεπερασμένο σύνολο T από τερματικά σύμβολα
- Ένα πεπερασμένο σύνολο P από κανόνες παραγωγής
- Ένα αρχικό σύμβολο S

Έτσι, μια τυπική γραμματική συμβολίζεται ως $G(V, T, P, S)$. Η ιεραρχία, περιλαμβάνει σε αυξημένη σειρά πολυπλοκότητας, τις κανονικές γραμματικές, τις γραμματικές χωρίς συμφραζόμενα, τις γραμματικές με συμφραζόμενα και τέλος, τις γενικές γραμματικές.

Στις **κανονικές γραμματικές** (regular grammars), οι οποίες ονομάζονται και γραμματικές τύπου 3, η μορφή των κανόνων παραγωγής τους είναι δεξιογραμμικές (right-linear) ή αριστερογραμμικές (left-linear). Αν είναι δεξιογραμμικές, τότε:

$$W_1 \rightarrow aW_2 \text{ ή } W \rightarrow a$$

ενώ, αν είναι αριστερογραμμικές:

$$W_1 \rightarrow W_2a \text{ ή } W \rightarrow a$$

Στις κανονικές γραμματικές, το πρώτο μέλος του κανόνα παραγωγής αποτελείται μόνο από ένα μη τερματικό σύμβολο, ενώ το δεύτερο μέλος περιέχει μια ακολουθία τερματικών συμβόλων και ένα μη τερματικό σύμβολο στα αριστερά ή στα δεξιά, ανάλογα αν η γλώσσα είναι δεξιογραμμική ή αριστερογραμμική, αντίστοιχα. Τις κανονικές γραμματικές αναγνωρίζουν τα Πεπερασμένα Αυτόματα (Finite State Automata). Αυτή η κατηγορία γλωσσών αντιστοιχεί, όπως θα δούμε, στις κανονικές εκφράσεις (regular expressions), οι οποίες έχουν πολλές εφαρμογές τόσο στη Βιοπληροφορική όσο και στην ανάλυση κειμένου. Κανονικές γλώσσες χρησιμοποιούνται, επίσης, για να οριστεί η λεξικογραφική δομή των γλωσσών προγραμματισμού.

Στις **γραμματικές χωρίς συμφραζόμενα** (context free grammar), οι οποίες ονομάζονται και γραμματικές τύπου 2, η μορφή των κανόνων παραγωγής τους είναι:

$$W \rightarrow \beta$$

Εδώ, το β είναι συμβολοσειρά (string) αποτελούμενη από οποιαδήποτε τερματικά ή μη-τερματικά σύμβολα (χωρίς όμως να συμπεριλαμβάνεται η κενή συμβολοσειρά). Τα αυτόματα που αναγνωρίζουν γραμματικές χωρίς συμφραζόμενα είναι τα Αυτόματα Στοίβας (Push Down Automata). Γλώσσες χωρίς συμφραζόμενα, αποτελούν τη θεωρητική βάση για τη δομή των φράσεων των περισσότερων γλωσσών προγραμματισμού παρόλο που το συντακτικό τους περιλαμβάνει και άλλα χαρακτηριστικά.

Στις **γραμματικές με συμφραζόμενα** (context sensitive grammar), οι οποίες ονομάζονται και γραμματικές τύπου 1, ανήκουν οι μονοτονικές γραμματικές (monotonic grammar). Η μορφή των κανόνων παραγωγής είναι:

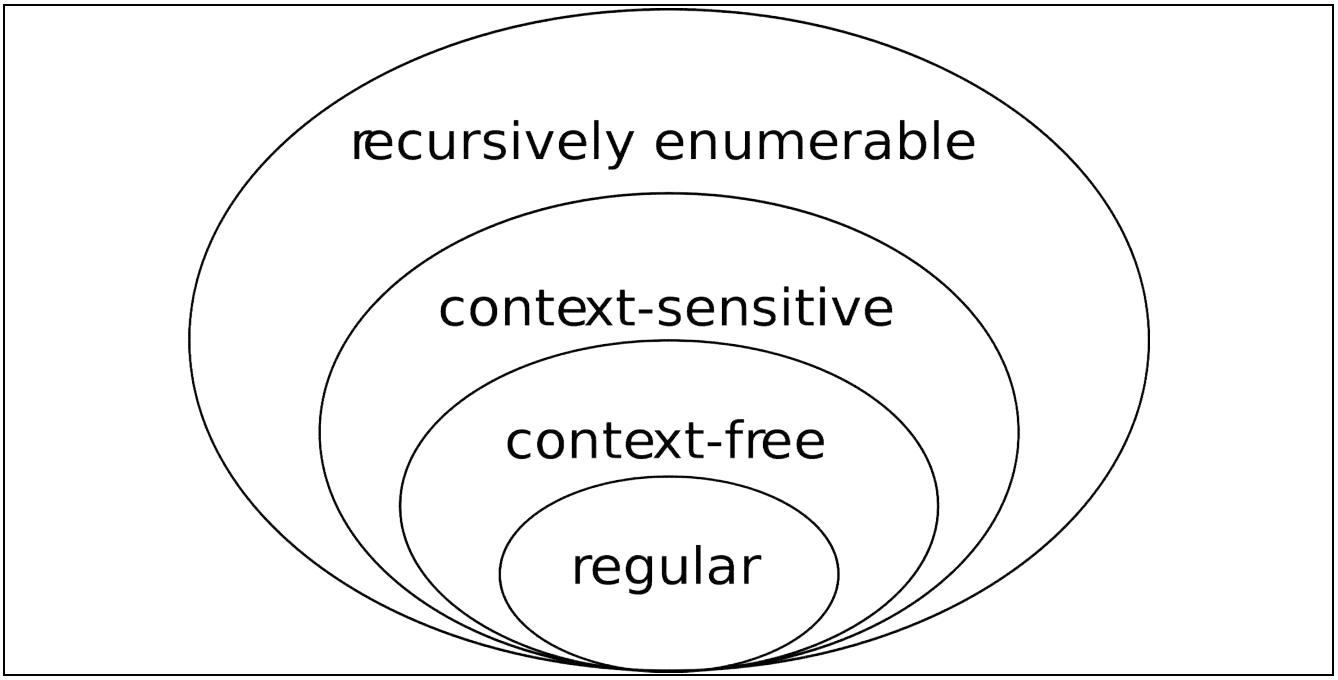
$$\alpha_1 W \alpha_2 \rightarrow a_1 \beta a_2$$

Εδώ, το a είναι ένα οποιοδήποτε τερματικό σύμβολο, το α οποιοσδήποτε συνδυασμός τερματικών και μη-τερματικών συμβόλων που περιλαμβάνει και την κενή συμβολοσειρά, ενώ το β οποιοσδήποτε συνδυασμός τερματικών και μη-τερματικών συμβόλων που δεν περιλαμβάνει και την κενή συμβολοσειρά. Παράγονται, έτσι, συμβολοσειρές μικρότερου μήκους από αυτό της αρχικής συμβολοσειράς. Γι' αυτό άλλωστε οι γλώσσες αυτές ονομάζονται μονοτονικές. Τα αυτόματα που αναγνωρίζουν γραμματικές χωρίς συμφραζόμενα είναι τα Γραμμικά Περιορισμένα Αυτόματα (Linearly Bounded Automata).

Τέλος, στις **γενικές γραμματικές** (unrestricted grammars), οι οποίες ονομάζονται και γραμματικές τύπου 0, η μορφή των κανόνων παραγωγής είναι:

$$\alpha_1 W \alpha_2 \rightarrow \beta$$

όπου β είναι οποιοσδήποτε συνδυασμός τερματικών και μη-τερματικών συμβόλων που περιλαμβάνει και την κενή συμβολοσειρά. Σε αυτήν την περίπτωση, οι συμβολοσειρές των κανόνων παραγωγής μπορούν να αποτελούνται από οποιαδήποτε σύμβολα της αλφαβήτου της γλώσσας. Από μια οποιαδήποτε συμβολοσειρά (εκτός της κενής) μπορεί να παραχθεί οποιαδήποτε άλλη (ή και η ίδια) συμβολοσειρά. Οι γενικές γραμματικές είναι γραμματικές με μόνο περιορισμό ότι από το κενό σύμβολο δεν παράγεται συμβολοσειρά. Επειδή δεν υπάρχουν άλλοι περιορισμοί, το σύνολο των γλωσσών που ανήκουν στις γενικές γραμματικές είναι το πιο ευρύ (συγκριτικά με τις υπόλοιπες γραμματικές της Ιεραρχίας του Τσόμσκι) και μέσα σε αυτό εμπεριέχονται τα σύνολα των γλωσσών που ανήκουν στις γραμματικές χαμηλότερης ιεραρχίας. Αυτές οι γλώσσες ονομάζονται και Αναδρομικώς Απαριθμήσιμες Γλώσσες (recursively enumerable languages). Οι γενικές γραμματικές αναγνωρίζονται από τις Μηχανές Τούρινγκ (Turing Machines).



Εικόνα 10.1: *Διαγραμματική απεικόνιση της ιεραρχίας των γραμματικών του Τσόμσκι. Κάθε ανώτερη γραμματική περιλαμβάνει σαν ειδική περίπτωση αυτές που βρίσκονται σε κατώτερο επίπεδο (από https://en.wikipedia.org/wiki/Chomsky_hierarchy).*

10.2. Κανονικές γραμματικές

Οι κανονικές γραμματικές, είδαμε ότι είναι γραμμικές, είτε δεξιογραμμικές είτε αριστερογραμμικές. Ας θεωρήσουμε μια απλή δεξιογραμμική γραμματική με αλφάβητο τα σύμβολα x και y, και κανόνες $S \rightarrow xS$ και $S \rightarrow y$. Αυτή η γραμματική μπορεί να παράξει όλες τις συμβολοσειρές που ξεκινάνε με αυθαίρετο αριθμό x και τελειώνουν με ένα μόνο y. Μπορεί να παράγει για παράδειγμα μια ακολουθία $S \rightarrow xS \rightarrow xxS \rightarrow xxxS \rightarrow xxxy$, όπου το βέλος συμβολίζει την εφαρμογή του κανόνα (μερικοί συγγραφείς συμβολίζουν την εφαρμογή του κανόνα παραγωγής με το διπλό βέλος $\Rightarrow$). Σε αυτή την περίπτωση έχουμε τρεις διαδοχικές εφαρμογές του πρώτου κανόνα και μια εφαρμογή του δεύτερου, έτσι ώστε να παραχθεί τελικά η συγκεκριμένη συμβολοσειρά, μία από τις άπειρες συμβολοσειρές που μπορούν να παραχθούν από αυτή τη γλώσσα.

Όπως είπαμε, τις κανονικές γραμματικές τις αναγνωρίζουν, δηλαδή τις διαβάζουν (parsing) τα Πεπερασμένα Αυτόματα (Finite State Automata). Αυτή η κατηγορία γλωσσών αντιστοιχεί όπως θα δούμε στις κανονικές εκφράσεις (regular expressions) οι οποίες έχουν πολλές εφαρμογές τόσο στη Βιοπληροφορική όσο και στην ανάλυση κειμένου. Ας θεωρήσουμε μια πολλαπλή στοίχιση, όπως αυτή της Εικόνας 10.2, και μια κανονική έκφραση ή ένα πρότυπο της PROSITE που να την περιγράφουν (όπως είδαμε, οι κανονικές εκφράσεις και οι εκφράσεις της PROSITE είναι ισοδύναμες).

```
RU1A_HUMAN    SRSLKMRGQAFVIFKEVSSAT
SXLF_DROME    KLTGRPRGVAFVRYNKREEAQ
ROC_HUMAN     VGCSVHKGFAFVQYVNERNAR
ELAV_DROME    GNDTQTKGVGFIRFDKREEAT
```

Εικόνα 10.2: *Ένα παράδειγμα πολλαπλής στοίχισης.*

Στη συγκεκριμένη περίπτωση, το πρότυπο της PROSITE θα δίνεται από την έκφραση:

```
[RK]-G-{EDRKHPCG}-[AGSCI]-[FY]-[LIVA]-x-[FYM]
```

ενώ η αντίστοιχη κανονική έκφραση θα ήταν:

```
[RK]G[^EDRKHPCG][AGSCI][FY][LIVA].[FYM]
```

Η μετατροπή αυτών των εκφράσεων σε μια κανονική γραμματική, γίνεται αν γράψουμε έναν ξεχωριστό κανόνα για κάθε θέση της παραπάνω έκφρασης. Έτσι, στην πρώτη θέση θα έχουμε έναν κανόνα

που θα λέει ότι μετά την έναρξη (S) το πρώτο σύμβολο θα είναι R, και έναν άλλον κανόνα που θα λέει ότι το πρώτο σύμβολο μπορεί να είναι G. Αυτοί οι δύο κανόνες, συμπτύσσονται σε έναν που λέει ότι το πρώτο σύμβολο θα είναι R ή G. Στη συνέχεια προχωράμε στον επόμενο κανόνα για τη δεύτερη θέση, κ.ο.κ. Σε αυτό το παράδειγμα, αλλά και σε όλα τα παρακάτω, για να είμαστε σύμφωνοι με τους κανόνες των γραμματικών που λένε ότι για τα τερματικά σύμβολα χρησιμοποιούνται πεζά γράμματα, θα χρησιμοποιούμε r αντί για R, κ.ο.κ. Τελικά, το σύνολο των κανόνων που περιγράφει αυτή την κανονική έκφραση, θα είναι:

$$S \rightarrow rW_1|kW_1$$
$$W_1 \rightarrow gW_2$$
$$W_2 \rightarrow [afilmnqrstvwy]\ W3$$
$$W_3 \rightarrow [agsci]\ W_4$$
$$W_4 \rightarrow fW_5|yW_5$$
$$W_5 \rightarrow lW_6|iW_6|vW_6|aW_6$$
$$W_6 \rightarrow [acdefghiklmnpqrstvwy]\ W_7$$
$$W_7 \rightarrow f|y|m$$

Μια αλληλουχία αμινοξέων που συμφωνεί με αυτή τη γραμματική, δηλαδή συμφωνεί με την παραπάνω κανονική έκφραση, θα παραχθεί με διαδοχική εφαρμογή των κανόνων:

$$S \rightarrow rW_1$$
$$\rightarrow rgW_2$$
$$\rightarrow rgaW_3$$
$$\rightarrow rgacW_4$$
$$\rightarrow rgacfW_5$$
$$\rightarrow rgacfvW_6$$
$$\rightarrow rgacfvkW_7$$
$$\rightarrow rgacfvky$$

Όλες οι κανονικές εκφράσεις, και κατά συνέπεια όλα τα πρότυπα της PROSITE μπορούν να περιγραφούν με όρους μιας τέτοιας κανονικής γραμματικής. Για την ακρίβεια, οι αλγόριθμοι που κάνουν αναζήτηση τέτοιων εκφράσεων βασίζονται στη θεωρία των κανονικών γραμματικών και των πεπερασμένων αυτομάτων. Όπως θα παρατηρήσατε ήδη από το πρώτο βήμα του παραδείγματος, εμφανίζεται πάλι το δίλημμα «ποιο είναι πιο πιθανό; Το *r* ή το *g*;». Προφανώς, όπως είδαμε και στην περίπτωση των κανονικών εκφράσεων, μια τέτοια διάκριση δεν μπορεί να γίνει και όλες οι (πεπερασμένες) αλληλουχίες που παράγονται από αυτή τη γλώσσα είναι το ίδιο πιθανές. Μια αλληλουχία, όπως η *rgacfvky* ή η *kgacfvky*, απλά ταιριάζει στο μοντέλο, ενώ μια άλλη όπως η *agacfvky* απλά δεν ταιριάζει.

Για να μπορέσουμε να κάνουμε τη διάκριση ανάμεσα στα τερματικά σύμβολα με τη μεγαλύτερη πιθανότητα, από αυτά με τη μικρότερη θα πρέπει να κάνουμε, όμοια με την περίπτωση των προτύπων, την εισαγωγή των στοχαστικών γραμματικών. Όπως είδαμε στην περίπτωση των προτύπων, μια πρώτη γενίκευση είναι η περίπτωση των προφίλ, ενώ η πιο γενική μορφή είναι τα HMM. Ας θεωρήσουμε το πιο απλό HMM που είχαμε δει στο κεφάλαιο 8, ένα μοντέλο με δύο μόνο καταστάσεις (M + και M-) και 4 σύμβολα (αναφερόμαστε στο DNA). Το μοντέλο αυτό απεικονίζεται στην Εικόνα 10.3 και όπως θα δούμε, μπορεί με τους κατάλληλους ορισμούς να μετασχηματιστεί σε μία εντελώς ισοδύναμη στοχαστική κανονική γραμματική, απλώς με την προσθήκη των κατάλληλων πιθανοτήτων. Δηλαδή, για μια στοχαστική κανονική γραμματική, χρειαζόμαστε τα τερματικά και τα μη-τερματικά σύμβολα, τους κανόνες παραγωγής (όπως στις κανονικές γραμματικές), αλλά και τις αντίστοιχες πιθανότητες.

Στην αρχή, θα πρέπει να μοντελοποιήσουμε τους κανόνες που αναφέρονται στις μεταβάσεις μεταξύ των καταστάσεων, δηλαδή μεταξύ των μη-τερματικών συμβόλων (έχουμε εδώ και καταστάσεις έναρξης και τερματισμού):

$$B \rightarrow M+|M-|E$$
$$M+ \rightarrow M+|M-|E$$
$$M- \rightarrow M+|M-|E$$

Επίσης, πρέπει να ορίσουμε τις πιθανές περιπτώσεις εμφάνισης συμβόλων από κάθε κατάσταση, χωρίς ακόμα να ορίσουμε την αντίστοιχη πιθανότητα:

M+: a|c|g|t

M-: a|c|g|t

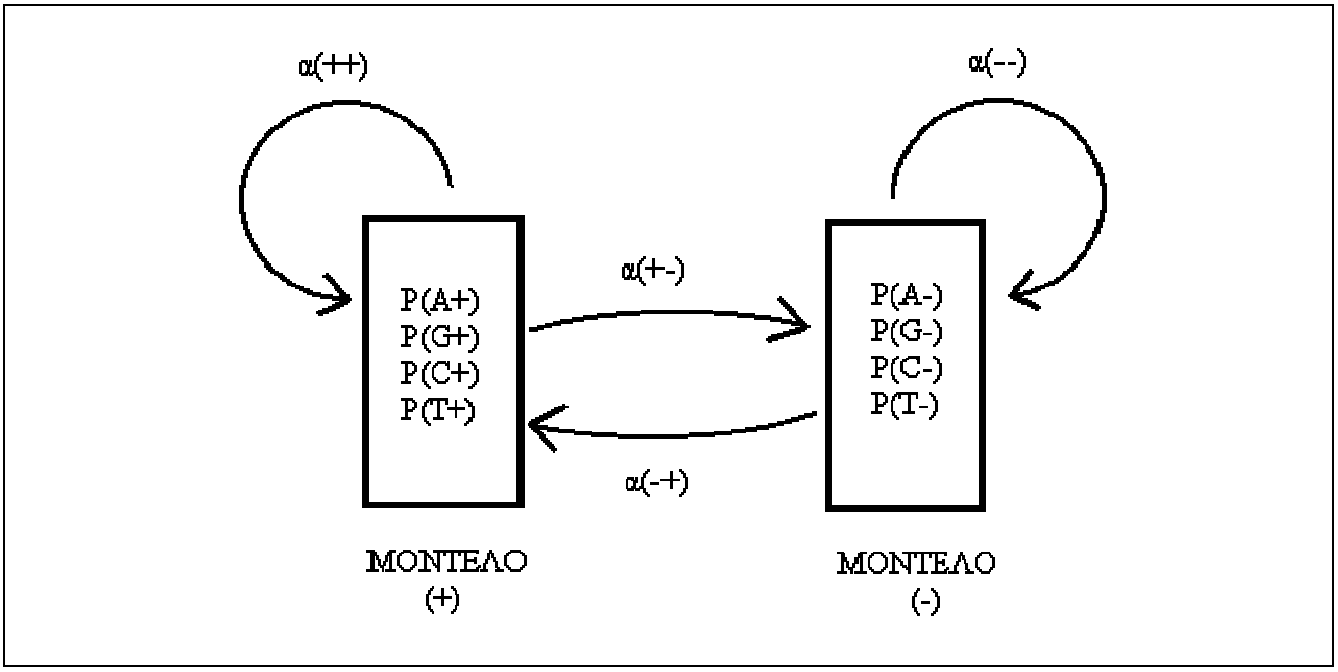
Σύμφωνα με την ορολογία των γραμματικών, αυτά είναι τα τερματικά σύμβολα. Έτσι για να ολοκληρωθεί το μοντέλο, πρέπει να συνδυαστούν τα παραπάνω υπολογίζοντας όλες τις πιθανές περιπτώσεις:

B → aM+|cM+|gM+|tM+|aM-|cM-|gM-|tM-|E

M+ → aM+|cM+|gM+|tM+|aM-|cM-|gM-|tM-|E

M- → aM+|cM+|gM+|tM+|aM-|cM-|gM-|tM-|E

Και τέλος, σε όλους αυτούς τους κανόνες, θα πρέπει να αντιστοιχήσουμε μια κατάλληλα υπολογισμένη πιθανότητα. Για παράδειγμα, για τον κανόνα $B \rightarrow aM+$ πρέπει να ορίσουμε την αντίστοιχη πιθανότητα ως $P(B \rightarrow aM+) = P(M+|B)P(a|M+)$, για τον κανόνα $M+ \rightarrow aM-$ την πιθανότητα $P(M+ \rightarrow aM-) = P(M-|M+)P(a|M-)$, κ.ο.κ.



Εικόνα 10.3: Ένα HMM με δύο καταστάσεις που έχουμε ήδη συναντήσει στο κεφάλαιο 8.

Ένα παράδειγμα παραγωγής (από τα άπειρα που μπορούν να υπάρξουν) από την παραπάνω γραμματική, είναι το:

$B \rightarrow aM-$
$\rightarrow aaM+$
$\rightarrow aacM+$
$\rightarrow aactM+$
$\rightarrow aactgM-$
$\rightarrow aactgcM-$
$\rightarrow aactgcaE$

Τα πεπερασμένα αυτόματα (Finite State Automata) που διαβάζουν τέτοιες γραμματικές είναι σε γενικές γραμμές οι μηχανές του Meale και οι μηχανές του Moore, οι οποίες αν και ορίζονται με διαφορετικό τρόπο, είναι σχεδόν ισοδύναμες μεταξύ τους (κάθε μηχανή του Moore μπορεί να μετατραπεί σε μια ισοδύναμη μηχανή του Meale, αλλά οι μηχανές του Meale δεν μπορούν όλες να μετατραπούν σε εντελώς ισοδύναμη μορφή). Γενικά πάντως, τέτοια αυτόματα παρόλο που χρησιμοποιούνται για κάποιες εφαρμογές στη μοντελοποίηση κυκλωμάτων, στη Βιοπληροφορική δεν έχουν εφαρμογές καθώς χρησιμοποιούνται οι πιο εύχρηστες δομές των κανονικών εκφράσεων και των HMM.

10.3. Γραμματικές χωρίς συμφραζόμενα και η πρόγνωση του RNA

Όπως είδαμε, οι κανονικές γραμματικές έχουν μια συγκεκριμένη κατεύθυνση στον τρόπο παραγωγής (αριστερά ή δεξιά). Κατά συνέπεια, κάποιες πιο σύνθετες δομές που απαντώνται στις βιολογικές αλληλουχίες (αλλά και σε άλλου είδους γλώσσες) δεν μπορούν να μοντελοποιηθούν με επάρκεια. Γραμματικές που έχουν τη δυνατότητα να παραθέσουν οποιονδήποτε συνδυασμό τερματικών και μη-τερματικών συμβόλων στο δεξί μέρος του κανόνα παραγωγής, έχουν μεγαλύτερη εκφραστική δύναμη και αποτελούν το επόμενο επίπεδο στην ιεραρχία του Chomsky. Οι γραμματικές αυτές όπως είδαμε, ονομάζονται γραμματικές χωρίς συμφραζόμενα (context-free grammars) και περιλαμβάνουν, για παράδειγμα, περιπτώσεις κατά τις οποίες μπορεί να παραχθεί ένας αριθμός συμβόλων ενός είδους ακολουθούμενος από ίσο αριθμό συμβόλων άλλου είδους (π.χ. η ακολουθία *xxxxyyyy*). Τέτοιες ακολουθίες δεν μπορούν να παραχθούν αμφιμονοσήμαντα (άρα, και να αναγνωριστούν στη συνέχεια) από μια κανονική γραμματική, γιατί τα πεπερασμένα αυτόματα δεν έχουν τρόπο να «θυμούνται» πόσες φορές έχει προηγηθεί ένα σύμβολο έτσι ώστε να παραχθούν άλλες τόσες φορές αντίγραφα από το άλλο σύμβολο. Στα HMM είδαμε ότι υπάρχουν τρόποι να μοντελοποιηθούν κάποιες ειδικές περιπτώσεις με τη χρήση πολλών διαφορετικών καταστάσεων, αλλά το γενικό πρόβλημα παραμένει.

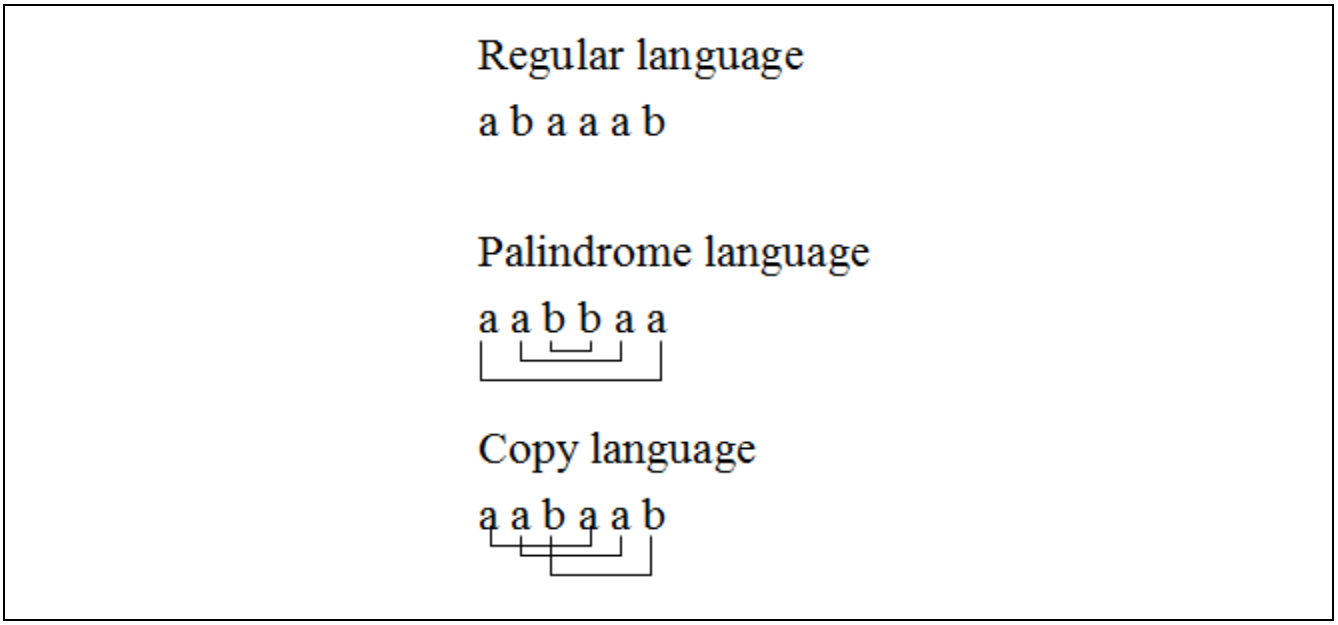
Το πρόβλημα αυτό το αντιμετωπίζουν πολύ εύκολα οι γραμματικές χωρίς συμφραζόμενα με το να επιτρέπουν κανόνες παραγωγής του είδους $S \rightarrow xSy$ οι οποίοι εξασφαλίζουν πάντα την ταυτόχρονη παραγωγή (αριστερά και δεξιά) ενός *x* και ενός *y*. Τα αυτόματα στοίβας που χαρακτηρίζουν αυτές τις γραμματικές, μπορούν να κρατήσουν τη «μνήμη» τέτοιων διαδοχικών καταστάσεων και έτσι η παραγωγή αλληλουχιών (συμβολοσειρών γενικότερα) που διαθέτουν τέτοιες εξαρτήσεις των τερματικών συμβόλων, είναι εφικτές, με την προϋπόθεση ότι οι εξαρτήσεις αυτές είναι είτε ανεξάρτητες μεταξύ τους (δηλαδή μεταξύ των ζευγαριών), είτε φωλιασμένες, αλλά ποτέ διασταυρούμενες. Το πιο χαρακτηριστικό παράδειγμα γλώσσας με την παραπάνω δομή, είναι η παλίνδρομη γλώσσα (Εικόνα 10.4). Τέτοια κείμενα, είναι γνωστά σαν καρκινικές επιγραφές και ονομάζονται οι συμμετρικές φράσεις οι οποίες μπορούν να διαβαστούν είτε από την αρχή είτε από το τέλος. Το κλασικότερο παράδειγμα από την Ελληνική ιστορία, είναι το «*ΝΙΨΟΝ ΑΝΟΜΗΜΑΤΑ ΜΗ ΜΟΝΑΝ ΟΨΙΝ*» η οποία χαρασσόταν συχνά σε πηγές και σε ελεύθερη μετάφραση στα νέα ελληνικά σημαίνει: «*πλύνε τις αμαρτίες, όχι μόνο το πρόσωπο*». Το πιο γνωστό σύγχρονο παράδειγμα τέτοιας τεχνητά κατασκευασμένης φράσης, αναφέρεται στην ερευνητική ομάδα του Bletchley Park, στην οποία συμμετείχε ο Alan Turing και είχε σκοπό (τον οποίο και πέτυχε τελικά) να σπάσει τον κώδικα του ENIGMA, της μηχανής κρυπτογράφησης που χρησιμοποιούσαν οι Γερμανοί στον Β' παγκόσμιο πόλεμο για να κωδικοποιούν τα μηνύματά τους. Οι κρυπτογράφοι αυτοί, είχαν σαν παιχνίδι να φτιάχνουν τέτοιες φράσεις και, όπως αναφέρουν, η πιο ωραία παλίνδρομη έκφραση που κατασκευάστηκε ποτέ, αποδίδεται στον Peter Hilton: "*Doc, note. I dissent. A fast never prevents fatness. I diet on cod*." (η φράση αυτή δεν είναι μόνο όμορφη γραμματικά, αλλά δίνει και σωστές...διατροφικές συμβουλές!).

Για παράδειγμα, η παλίνδρομη φράση aabaabaa μπορεί να παραχθεί από μια γραμματική με τους κανόνες

$$S \rightarrow aSa|bSb|aa|bb$$

ως εξής: $S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aabaabaa$. Βλέπουμε, ότι για κάθε παραγωγή a υπάρχει και ένα συμμετρικό a (και όμοια για τα b). Το μήκος που θα έχει η φράση καθορίζεται από το αν θα υπάρξουν πολλές επαναλήψεις του κανόνα ο οποίος περιέχει το μη-τερματικό σύμβολο (αν εμφανιστεί στο δεξί μέλος ο κανόνας που περιέχει μόνο τερματικά σύμβολα, η αλληλουχία τερματίζεται).

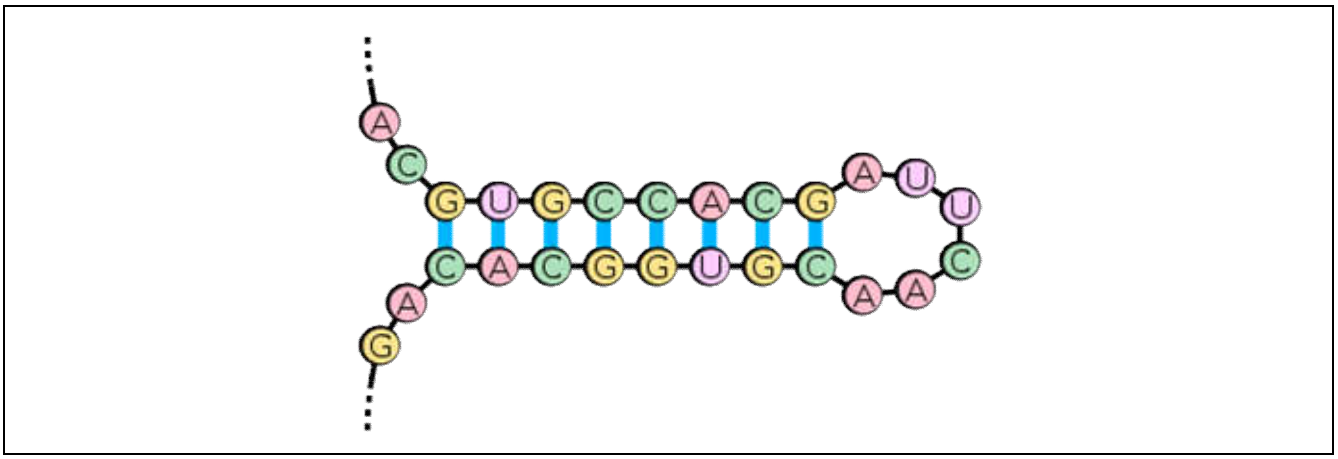
.



Εικόνα 10.4: *Παραδείγματα κανονικής γλώσσας, παλίνδρομης γλώσσας και αντιγραφικής γλώσσας.*

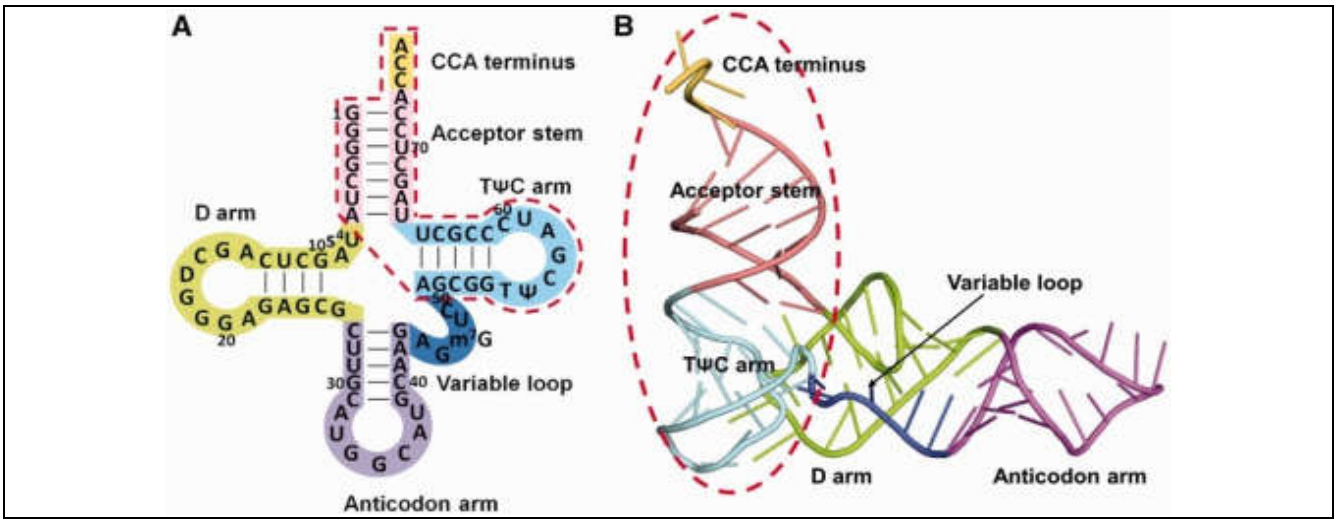
Στη βιολογία, ένα παράδειγμα χαρακτηριστικό της παλίνδρομης γλώσσας, είναι η δευτεροταγής δομή του RNA. Τα μόρια αυτά, σε αντίθεση με το DNA, αναδιπλώνονται και σχηματίζουν πολύπλοκες τρισδιάστατες δομές (παρόμοιες με των πρωτεϊνών), με σχηματισμό δεσμών υδρογόνου μεταξύ των συμπληρωματικών βάσεων (A-U, G-C). Έτσι, υπάρχει μια ξεκάθαρη «εξάρτηση» μεταξύ τμημάτων της αλληλουχίας που βρίσκονται μακριά το ένα από το άλλο. Στην πιο απλή μορφή, η εξάρτηση αυτή παίρνει τη μορφή φουρκέτας στην οποία τα απέναντι τοποθετημένα νουκλεοτίδια σχηματίζουν δεσμούς υδρογόνου. Καταλαβαίνουμε λοιπόν, ότι αν μπορούσαμε να μοντελοποιήσουμε κατάλληλα ένα τέτοιο σύστημα, θα μπορούσαμε να προβλέψουμε τη δευτεροταγή δομή του RNA και με τη χρήση αυτών των εξαρτήσεων να έχουμε μια πολύ καλή προσέγγιση για την τρισδιάστατη δομή του. Στην Εικόνα 10.5 δίνεται ένα παράδειγμα μια τέτοιας λούπας (loop) στην οποία φαίνονται και οι δεσμοί υδρογόνου μεταξύ των συμπληρωματικών βάσεων. Η δευτεροταγής δομή που προκύπτει για την περιγραφή ενός τέτοιου δομικού μοτίβου, περιγράφεται ως:

```
acgugccacgauucaacguggcacag
..(((((((((.....)))))))))..
```



Εικόνα 10.5: *Παράδειγμα ενός τυπικού βρόχου σε μόριο RNA (από https://en.wikipedia.org/wiki/Stem-loop).*

Στη δομή αυτή, οι εξαρτήσεις συμβολίζονται με τις παρενθέσεις, και βλέπουμε έτσι ότι η Γουανίνη στη θέση 3 κάνει δεσμό υδρογόνου με την Κυτοσίνη στη θέση 25, η Ουρακίλη στη θέση 3 με την Αδενίνη στη θέση 24 κ.ο.κ. Οι βάσεις οι οποίες δεν εμπλέκονται σε τέτοιες αλληλεπιδράσεις, συμβολίζονται με την τελεία (.).



Εικόνα 10.6: *Δευτεροταγής (Α) και τριτοταγής δομή (Β) του tRNAPhe (PDB code 6TNA) της E. coli,. (Ito et al., 2012)*

Όπως γίνεται φανερό, για να πετύχουμε την παραπάνω αναπαράσταση με μια γραμματική με συμφραζόμενα, αρκεί να ορίσουμε ρητά στους κανόνες ότι θα εξασφαλίζεται η συμμετρική κατανομή και εμφάνιση των τερματικών με τρόπο που να υπακούν στον κανόνα της συμπληρωματικότητας:

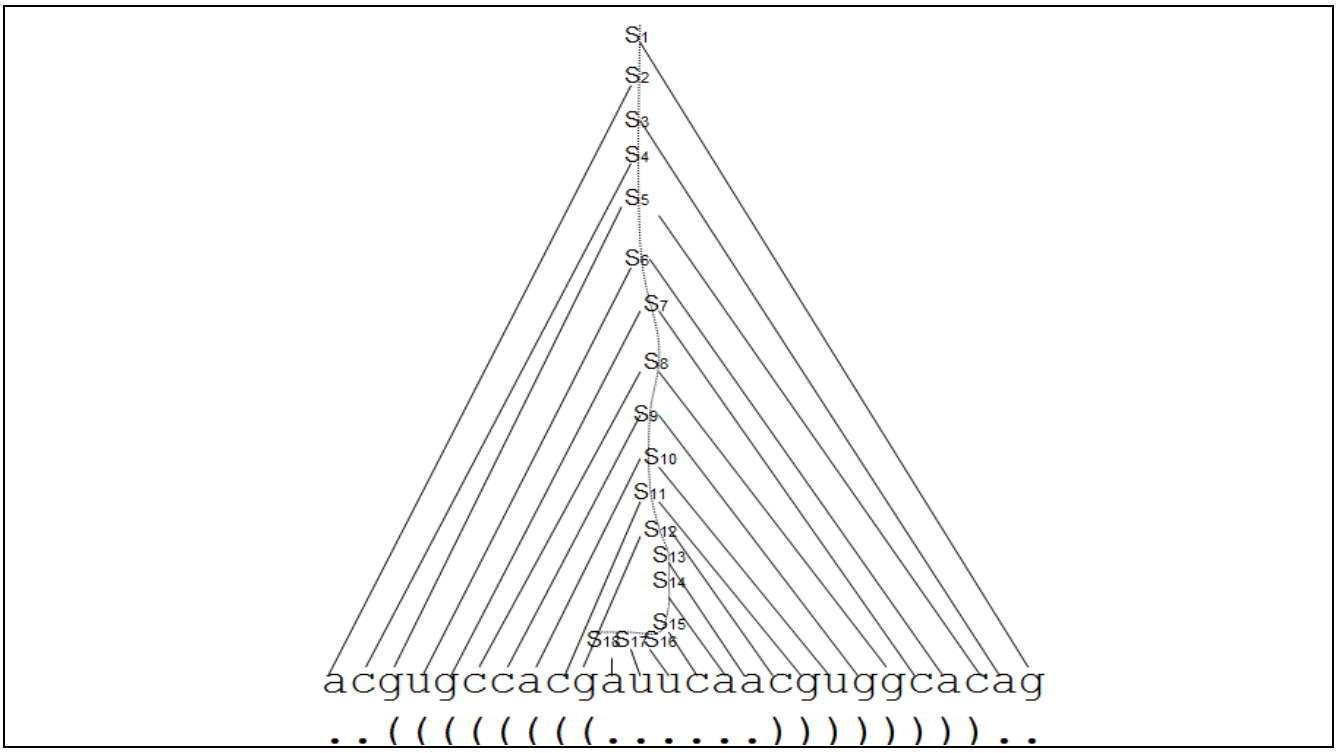
$$S \rightarrow gSc|cSg|\ aSu|uSa$$

Προφανώς, για να μπορούμε να μοντελοποιήσουμε μια πραγματικά πολύπλοκη δομή, οι κανόνες πρέπει να είναι περισσότεροι και να διαδέχονται ο ένας τον άλλον, ενώ θα πρέπει να εξασφαλίσουμε και το τι θα γίνεται στις περιπτώσεις που έχουμε βάσεις που δε συμμετέχουν σε δεσμούς υδρογόνου (π.χ. θα πρέπει να υπάρχουν και κανόνες του τύπου $S \rightarrow gS|\ Sg$), τι θα γίνεται στην περίπτωση περισσότερων βρόχων (θα πρέπει να υπάρχουν διακλαδώσεις που εξασφαλίζονται από κανόνες του τύπου $S \rightarrow S_1S_2$) και πώς θα τερματίζεται η αλληλουχία (θα πρέπει να υπάρχουν κανόνες του τύπου $S \rightarrow g$).

$S_1 \rightarrow S_2g$ $\quad$ $S_{10} \rightarrow aS_{11}u$
$S_2 \rightarrow aS_3$ $\quad$ $S_{11} \rightarrow cS_{12}g$
$S_3 \rightarrow S_4a$ $\quad$ $S_{12} \rightarrow gS_{13}c$
$S_4 \rightarrow aS_5$ $\quad$ $S_{13} \rightarrow aS_{14}$
$S_5 \rightarrow gS_6c$ $\quad$ $S_{14} \rightarrow uS_{15}$
$S_6 \rightarrow uS_7a$ $\quad$ $S_{15} \rightarrow uS_{16}$
$S_7 \rightarrow gS_8c$ $\quad$ $S_{16} \rightarrow cS_{17}$
$S_8 \rightarrow cS_9g$ $\quad$ $S_{17} \rightarrow aS_{18}$
$S_9 \rightarrow cS_{10}g$ $\quad$ $S_{18} \rightarrow a$

$$S_1 \rightarrow S_2g$$
$$\rightarrow aS_3g$$
$$\rightarrow aS_4ag$$
$$\rightarrow aaS_5ag$$
$$\rightarrow aagS_6cag$$
$$\rightarrow aaguS_7acag$$
$$\rightarrow aagugS_8cacag$$
$$\rightarrow aagugcS_9gcacag$$
$$\rightarrow aagugccS_{10}ggcacag$$
$$\rightarrow aagugccaS_{11}uggcacag$$
$$\rightarrow aagugccacS_{12}guggcacag$$
$$\rightarrow aagugccacgS_{13}cguggcacag$$
$$\rightarrow aagugccacgaS_{14}cguggcacag$$
$$\rightarrow aagugccacgauS_{15}cguggcacag$$

$\rightarrow$aagugccacgauuS_{16}cguggcacag
$\rightarrow$aagugccacgauucS_{17}cguggcacag
$\rightarrow$aagugccacgauucaS_{18}cguggcacag
$\rightarrow$aagugccacgauucaacguggcacag



Εικόνα 10.7: *Η παραγωγή της δευτεροταγούς δομής του RNA σε αναπαράσταση δέντρου.*

Αυτό που γίνεται βέβαια κατανοητό για τη γλώσσα του παραπάνω παραδείγματος είναι ότι, είναι πολύ ειδική, μπορεί να παράγει δηλαδή μόνο τη συγκεκριμένη αλληλουχία RNA. Σε μια πραγματική περίσταση θα έπρεπε οι κανόνες να είναι πολύ περισσότεροι και πιο γενικοί, έτσι ώστε να μπορούν να παραχθούν από αυτούς, αλλά και το κυριότερο, να μπορούν να αναγνωριστούν, περισσότερα μόρια μιας συγκεκριμένης κατηγορίας (πχ tRNA).

Το επόμενο λογικό βήμα, θα είναι οι παραπάνω γραμματικές, να γίνουν στοχαστικές. Αυτό επιτυγχάνεται, όπως και στην περίπτωση των κανονικών γραμματικών, με την προσθήκη μιας κατάλληλης πιθανότητας σε κάθε κανόνα. Η διαδικασία αυτή οδηγεί στις πολύ γνωστές «στοχαστικές γραμματικές χωρίς συμφραζόμενα» (stochastic context-free grammars). Το βασικό πλεονέκτημα που έχουν αυτά τα μοντέλα, είναι το αντίστοιχο που είχαν και οι στοχαστικές κανονικές γραμματικές έναντι των κανονικών γραμματικών, ή τα HMM έναντι των κανονικών εκφράσεων: η επέκταση και εκλέπτυνση των αποτελεσμάτων και η ενσωμάτωση των περιπτώσεων με μικρή πιθανότητα εμφάνισης (με αντίστοιχη ποσοτικοποίηση). Ένα κλασικό παράδειγμα στην περίπτωση του RNA είναι το ότι μπορεί πλέον με τη χρήση (μικρών) πιθανοτήτων να επιτρέπουμε το «λαθεμένο» ζευγάρωμα βάσεων, G-U, C-A, κάτι που πιθανώς να δώσει πιο ρεαλιστικές προβλέψεις.

Κατ' αντιστοιχία με τα HMM, στις στοχαστικές γραμματικές χωρίς συμφραζόμενα, έχουν εφαρμογή τα τρία κλασικά ερωτήματα:

- Πώς θα επιτύχουμε την καλύτερη στοίχιση μιας ακολουθίας με μια γραμματική (alignment-parsing problem)
- Πώς θα υπολογίσουμε την πιθανότητα μιας ακολουθίας δεδομένης μιας γραμματικής (scoring problem)
- Πώς θα γίνει η εύρεση των βέλτιστων παραμέτρων μιας γραμματικής αν υπάρχουν γνωστά παραδείγματα (training problem)

Παρόλο που λόγω πολυπλοκότητας και χώρου δεν θα μπούμε σε λεπτομέρειες, οι απαντήσεις στα προβλήματα αυτά ακολουθούν επίσης μια πορεία ανάλογη με αυτήν των HMM. Το πρώτο ερώτημα, αφορά την εφαρμογή του αλγόριθμου του Viterbi στις γραμματικές. Ο αλγόριθμος αυτός ονομάζεται αλγόριθμος των Cocke-Younger-Kasami (CYK algorithm) και πρώτη φορά προτάθηκε από τον Youmger το 1967 (Younger, 1967). Στο δεύτερο ερώτημα, ο αντίστοιχος αλγόριθμος είναι ο αλγόριθμος Inside (outside algorithm) που είναι αντίστοιχος του Forward (ενώ ο outside είναι ο αντίστοιχος του Backward). Τέλος, το συνολικό πρόβλημα της εκπαίδευσης, απαντάται από τον αλγόριθμο Inside-Outside ο οποίος είναι αντίστοιχος του αλγορίθμου Baum-Welch (Forward-Backward) και προτάθηκε το 1979 (Baker, 1979). Όπως είναι φανερό, οι αλγόριθμοι αυτοί, σε αντίθεση με τους αλγόριθμους των HMM οι οποίοι αντιμετώπιζαν την αλληλουχία σειριακά, θα πρέπει να αντιμετωπίσουν την αλληλουχία κάνοντας χρήση του δέντρου, γι' αυτό προκύπτει και το όνομα (inside/outside). Επόμενο είναι λοιπόν, όλα τα παραπάνω να αποτυπώνονται και στην αλγοριθμική πολυπλοκότητα και στις απαιτήσεις σε μνήμη αυτών των αλγορίθμων (Πίνακας 10.1).

Μια άλλη σημαντική αναφορά, πρέπει να γίνει στη λεγόμενη «*κανονική μορφή του Chomsky*» (Chomsky Normal Form). Ο Chomsky πρότεινε το 1959 (Chomsky, 1959) ότι κάθε γραμματική χωρίς συμφραζόμενα, μπορεί να γραφτεί κάνοντας χρήση μόνο κανόνων όπως:

$$W_1 \rightarrow W_2 W_3 \text{ ή}$$
$$W_1 \rightarrow a$$

Επίσης, ισχύει και το αντίστροφο, δηλαδή κάθε γραμματική που παίρνει αυτή τη μορφή, είναι υποχρεωτικά γραμματική χωρίς συμφραζόμενα. Όταν μετατρέπουμε μια γραμματική χωρίς συμφραζόμενα στην κανονική μορφή Chomsky το μέγεθος της νέας γραμματικής αναγκαστικά θα μεγαλώνει, αλλά δεν μπορεί να είναι μεγαλύτερο από το τετράγωνο του μεγέθους της αρχικής γραμματικής (όπου «μέγεθος» εννοούμε τον αριθμό των κανόνων). Προφανώς, η μετατροπή πολύπλοκων γραμματικών δεν είναι απλή υπόθεση και έχουν προταθεί αλγόριθμοι με προκαθορισμένα βήματα για το σκοπό αυτό (Lange & Leiß, 2009). Το μεγάλο πλεονέκτημα από τη χρήση της κανονικής μορφής, από όπου προκύπτει και η σημασία της, εντοπίζεται στη χρησιμότητά της στους υπολογισμούς και στους αλγόριθμους, καθώς με τη μορφή αυτή διευκολύνονται πολύ οι υπολογισμοί και η υλοποίηση.

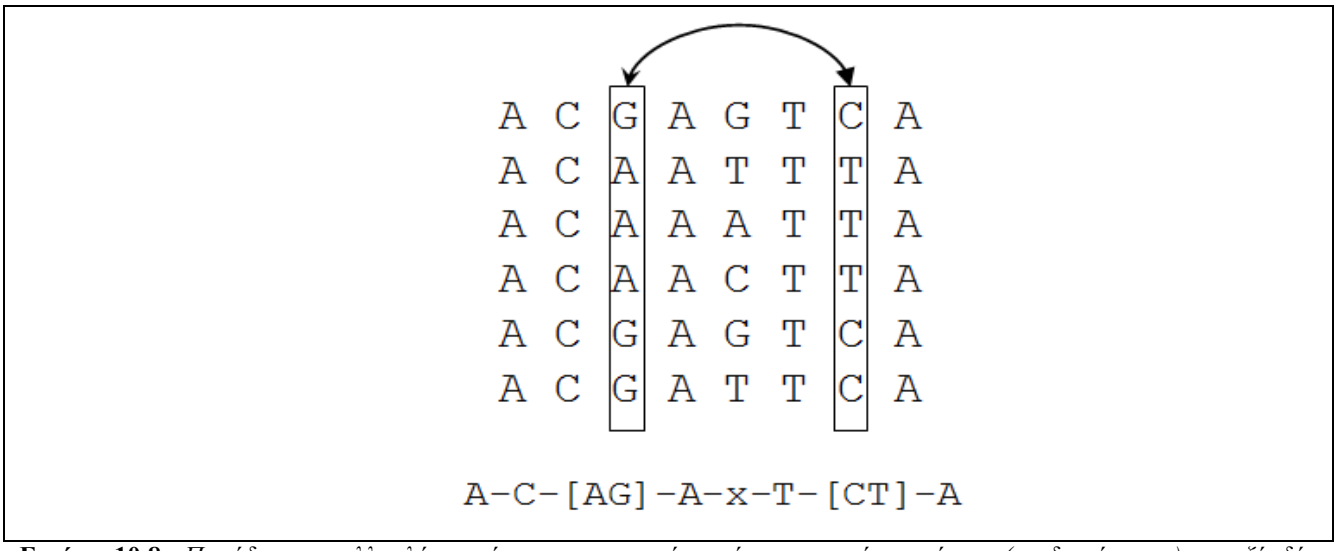
Στόχος	**HMM**	**SCFG**
Βέλτιστη στοίχιση	Viterbi	CYK
$P(\mathbf{x}\|\theta)$	Forward	Inside
EM algorithm	Baum-Welch	Inside-Outside
Απαιτήσεις σε μνήμη	$O(LM)$	$O(L^2M)$
Πολυπλοκότητα	$O(LM^2)$	$O(L^3M^3)$

Πίνακας 10.1: *Αντιστοίχιση των εννοιών μεταξύ HMM και SCFG.*

Οι πρώτες εφαρμογές των στοχαστικών γραμματικών με συμφραζόμενα στη μελέτη του RNA έγιναν τη δεκαετία του 1990 από τον Sakakibara (Sakakibara et al., 1994). Αξίζει να σημειωθεί, ότι μέχρι τότε οι πιο επιτυχημένες προσπάθειες πρόγνωσης των RNA βασίζονταν στις εργασίες της Nussinov και του Zuker. Η Nussinov είχε παρουσιάσει πρώτη το 1978 έναν κομψό αλγόριθμο δυναμικού προγραμματισμού ο οποίος μεγιστοποιούσε το σύνολο των ζευγαριών βάσεων που βρίσκονταν σε δίκλωνη μορφή (Nussinov, Pieczenik, Griggs, & Kleitman, 1978). Ο Zuker παρουσίασε λίγα χρόνια αργότερα έναν αλγόριθμο βασισμένο στη θερμοδυναμική, ο οποίος μεγιστοποιούσε μια συνάρτηση ελεύθερης ενέργειας (ΔG). Με τη μέθοδο αυτή λαμβάνεται υπόψη η συνολική ενεργειακή κατάσταση του μορίου, και πιθανώς να επιτρέπεται και το «λαθεμένο» ζευγάρωμα βάσεων, G-U, C-A και κατά συνέπεια, η μέθοδος αυτή αποδίδει καλύτερα (Zuker & Stiegler, 1981). Και οι δυο αλγόριθμοι, βρέθηκε αργότερα ότι μπορούν να γραφούν σε μια ισοδύναμη μορφή SCFG, αλλά σε γενικές γραμμές οι μεθοδολογίες που βασίζονται σε θερμοδυναμικούς υπολογισμούς ελεύθερης ενέργειας εξακολουθούν να είναι ιδιαίτερα ακριβείς, κυρίως λόγω των ευριστικών τεχνικών που ενσωματώνουν. Μια απλή εξήγηση των αλγορίθμων δυναμικού προγραμματισμού, παραθέτει ο Eddy (Sean R Eddy, 2004). Στη μέθοδο του Zuker βασίζεται η πολύ γνωστή μέθοδος **MFOLD** (http://unafold.rna.albany.edu/?q=mfold), η οποία είναι ίσως και μια από τις παλιότερες διαδικτυακές εφαρμογές στη Βιοπληροφορική. Το **RNAfold** (http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi) είναι επίσης μια πολύ γνωστή εφαρμογή, που χρησιμοποιεί μεταξύ άλλων, και τον αλγόριθμο του Zuker για την πρόγνωση των RNA (Lorenz et al., 2011). Το **PFOLD** (http://www.daimi.au.dk/~compbio/pfold) είναι ίσως η πιο επιτυχημένη εφαρμογή για πρόγνωση δομής RNA και βασίζεται σε γραμματικές χωρίς συμφραζόμενα (Knudsen & Hein, 2003). Οι Dowell και Eddy (Dowell & Eddy, 2004) πραγματοποίησαν μια μεγάλη

συγκριτική μελέτη στην οποία υλοποίησαν μια σειρά από διαφορετικές γραμματικές χωρίς συμφραζόμενα, ειδικά για την περίπτωση της πρόγνωσης της δομής του RNA. Μελέτησαν τις διαφορές των διαφόρων γραμματικών και πραγματοποίησαν συγκρίσεις έναντι των κλασικών αλγορίθμων ελαχιστοποίησης ενέργειας. Τα αποτελέσματα έδειξαν ότι κάποιες από τις γραμματικές αυτές, μπορούσαν να δώσουν αποτελέσματα συγκρίσιμα με τους κλασικούς αλγορίθμους, ενώ ο αλγόριθμος του PFOLD ήταν και ιδιαίτερα φειδωλός (και άρα και γρήγορος). Η μελέτη αυτή έχει και μια ιδιαίτερη σημασία καθώς ο κώδικας των γραμματικών αυτών, το λογισμικό **CONUS**, είναι διαθέσιμος, για μελλοντική χρήση και πειραματισμούς (http://selab.janelia.org/software/conus/). Παρόμοιο αποτελέσματα έδειξε και μια μεταγενέστερη μελέτη με χρήση του λογισμικού **TORNADO** το οποίο δίνει περισσότερες δυνατότητες μοντελοποίησης και εφαρμογής σε άλλες περιπτώσεις (http://selab.janelia.org/software/tornado/tornado.tar.gz) (Rivas, Lang, & Eddy, 2012).

Μια άλλη πολύ σημαντική εφαρμογή των γραμματικών αυτών στη μελέτη των RNA, ξεκίνησε από την δουλειά των Eddy και Durbin, σε μια κατηγορία μοντέλων που ονομάζονται Covariance Models (μοντέλα συνδυακύμανσης) (Eddy & Durbin, 1994). Τα μοντέλα αυτά, είναι μια ειδική περίπτωση γραμματικών χωρίς συμφραζόμενα, κατάλληλα φτιαγμένα για να περιγράφουν μια πολλαπλή στοίχιση RNA. Τα μοντέλα αυτά ειναι για τα SCFG, ό,τι είναι τα profile HMM για τα HMM. Η διαφορά τους από τα γενικά μοντέλα, είναι ότι είναι φτιαγμένα για να λειτουργούν πάνω στην πολλαπλή στοίχιση. Αυτό τους δίνει μεγαλύτερη ευελιξία από άποψη υπολογιστικής πολυπλοκότητας, αλλά καθιστά πιο δύσκολη την εφαρμογή τους σε περιπτώσεις που μια πολλαπλή στοίχιση μιας οικογένειας δεν είναι διαθέσιμη. Το όνομα βγαίνει από τη συνδιακύμανση των διαδοχικών στηλών μιας πολλαπλής στοίχισης. Η δομή της γραμματικής, επιτρέπει τη μοντελοποίηση μιας τέτοιας αλληλεπίδρασης, η οποία δεν ήταν δυνατή με τα πρότυπα κανονικών εκφράσεων, αλλά ούτε και με τα HMM.



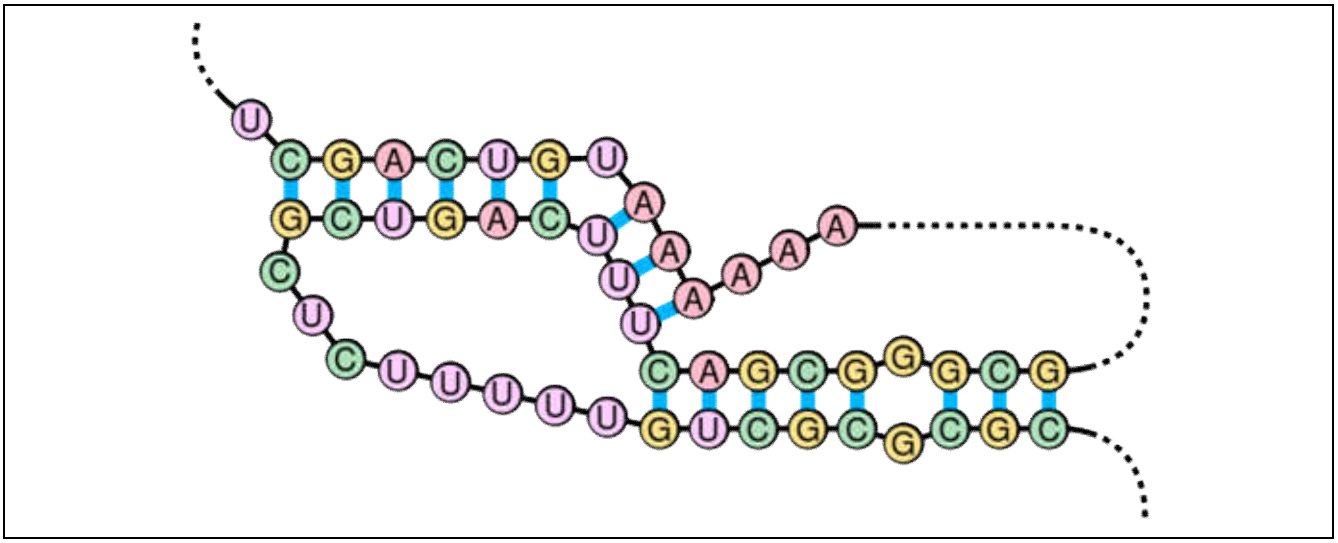
Εικόνα 10.8: *Παράδειγμα πολλαπλής στοίχισης στην οποία υπάρχει ισχυρή συσχέτιση (συνδιακύμανση) μεταξύ δύο στηλών.*

Μια πολλαπλή στοίχιση στην οποία υπάρχει ισχυρή συνδιακύμανση (δηλαδή, αλληλεπίδραση) μεταξύ της στήλης 3 και της στήλης 7, φαίνεται στην Εικόνα 10.9. Αν εξετάσουμε κάθε στήλη ξεχωριστά, βλέπουμε ότι στην 3 έχουμε 50% G και 50% A, ενώ στην 7 έχουμε 50% T και 50% C. Το απλό HMM ή ένα πρότυπο PROSITE (το οποίο θα ήταν A-C-[AG]-A-x-T-[CT]-A), θα έδινε για παράδειγμα την ίδια πιθανότητα να εμφανιστεί G (3^{η}) και C(7^{η}), και στο να εμφανιστεί G(3^{η}) και T (7^{η}). Παρατήρηση όμως των συχνοτήτων των δινουκλεοτιδίων, μας δείχνει ότι όταν υπάρχει G (3^{η}) υπάρχει πάντα C(7^{η}), ενώ όταν υπάρχει A(3^{η}) πάντα ακολουθείται από T (7^{η}). Αυτή η εξάρτηση, μπορεί να αποτυπωθεί στο covariance model, και στο συγκεκριμένο παράδειγμα το μοντέλο αυτό θα έδινε πολύ μεγάλο σκορ σε αυτή την πολλαπλή στοίχιση και θα ταξινομούσε σε αυτή την κατηγορία μια αλληλουχία όπως την ACGATTCA, αλλά θα απέρριπτε την ακολουθία ACGATTTA. Και οι δύο όμως αλληλουχίες θα ταίριαζαν (λαθεμένα) με το παραπάνω πρότυπο PROSITE.

Στα μοντέλα συνδιακύμανσης, βασίζεται το γνωστό πακέτο λογισμικού **INFERNAL**, http://infernal.wustl.edu/ το οποίο έχει υλοποιήσει και συντηρεί ο Sean Eddy (Nawrocki, Kolbe, & Eddy, 2009), και παρουσιάζει πολλές ομοιότητες με το ήδη γνωστό πακέτο HMMER για τα HMM. Για την

ακρίβεια, το INFERNAL δεν προβλέπει δευτεροταγή δομή, αλλά βρίσκει αν ένα RNA ανήκει σε μια γνωστή οικογένεια, αν ταιριάζει σε μια δεδομένη πολλαπλή στοίχιση. Αν τώρα κάποιο μέλος της οικογένειας διαθέτει δομή, η πρόγνωση γίνεται έμμεσα. Φυσικά, ένα μεγάλο πλεονέκτημα του λογισμικού είναι η ευκολία στη χρήση και η δυνατότητα, ο χρήστης να κατασκευάσει μοντέλα για τις δικές του οικογένειες RNA. Κατ' αναλογία με τη βάση PFAM, η οποία περιέχει στοιχίσεις οικογενειών πρωτεϊνών, στο INFERNAL βασίζεται η βάση δεδομένων RFAM, η οποία περιέχει οικογένειες RNA, http://rfam.xfam.org/ (Gardner et al., 2011). Το **EvoFold**, http://users.soe.ucsc.edu/~jsp/EvoFold/, χρησιμοποιεί μια παρόμοια αλλά κάπως πιο προχωρημένη τεχνική για να περιγράψει τις πολλαπλές στοιχίσεις, η οποία βασίζεται σε φυλογενετική ανάλυση και εξελικτική πληροφορία (phylo-SCFG). Το πλεονέκτημα της μεθόδου είναι ότι μπορεί να χρησιμοποιηθεί και για άλλες κατηγορίες RNA όπως microRNA (Pedersen et al., 2006). Το **RNAz**, http://www.tbi.univie.ac.at/~wash/RNAz/ βασίζεται σε ένα συνδυασμό θερμοδυναμικών παραμέτρων και πολλαπλών στοιχίσεων που δείχνουν την εξελικτική πληροφορία (Washietl, Hofacker, & Stadler, 2005). Τέλος, το **CONTRAfold**, το οποίο είναι διαθέσιμο στη διεύθυνση http://contra.stanford.edu/contrafold, βασίζεται σε ένα κάπως διαφορετικό στοχαστικό μοντέλο το οποίο αποτελεί γενίκευση των SCFG και ανήκει στην κατηγορία των «διαχωριστικών» μοντέλων, και ονομάζεται conditional log-linear model (CLLM). Το CONTRAfold είναι από τις λίγες καθαρά πιθανοθεωρητικές μεθόδους που προσεγγίζει την ακρίβεια πρόγνωσης των θερμοδυναμικών μεθόδων (Do, Woods, & Batzoglou, 2006).

Πέραν όσων αναφέραμε παραπάνω, υπάρχουν περιπτώσεις ακόμα και στη δομή του RNA που ακόμα και οι γραμματικές χωρίς συμφραζόμενα δεν επαρκούν. Μια τέτοια επιπλοκή στην πρόγνωση δευτεροταγούς δομής του RNA έφερε η ανακάλυψη των ψευδοκόμπων (pseudoknots). Ψευδοκόμπος (Εικόνα 10.10) είναι μια δομή των νουκλεϊκών οξέων στην οποία οι βρόχοι διασταυρώνονται και διακλαδώνονται με συνέπεια η μία πλευρά (το στέλεχος) του ενός να κάνει δεσμούς υδρογόνου με τη μία πλευρά του άλλου. Όπως φαίνεται και από την Εικόνα 10.10, ο ψευδοκόμπος δεν μπορεί να αναπαρασταθεί από μια παλίνδρομη γλώσσα γιατί θα απαιτείται δέντρο με διασταυρούμενες συσχετίσεις, με άλλα λόγια οι κανόνες που παράγουν τα ζευγάρια βάσεων δεν μπορούν να είναι φωλιασμένοι (ο ένας μέσα στον άλλον). Για παράδειγμα, η αλληλουχία AAUUCCGG μπορεί να αναπαρασταθεί σαν δυο φωλιασμένες (nested) παλίνδρομες αλληλουχίες, αλλά η αλληλουχία AACCUUGG απαιτεί να υπάρχει διασταύρωση (crossing) των παλίνδρομων, πράγμα που δεν επιτρέπεται. Οι ψευδοκόμποι εισάγουν πολλά προβλήματα στους αλγορίθμους πρόγνωσης που αναφέραμε παραπάνω, τόσο στους κλασικούς αλγόριθμους δυναμικού προγραμματισμού, όσο και στις αντίστοιχες γραμματικές. Έχουν προταθεί ειδικοί αλγόριθμοι δυναμικού προγραμματισμού για να υπολογίζουν τη δομή σε μόρια με ψευδοκόμπους, αλλά υπάρχουν κάποια προβλήματα. Ένας ακριβής αλγόριθμος υπάρχει, αλλά μόνο για την περίπτωση που μεγιστοποιούμε απλά το σύνολο των ζευγαριών βάσεων (όπως ο αλγόριθμος της Nussinov), αλλά ακόμα και τότε η πολυπλοκότητά του είναι μεγάλη με συνέπεια να είναι αργός. Αν πάμε σε θερμοδυναμικούς υπολογισμούς, τότε έχει αποδειχθεί ότι το πρόβλημα είναι NP-complete (Lyngsø & Pedersen, 2000). Οι πιο συνηθισμένες περιπτώσεις, αφορούν αλγόριθμους δυναμικού προγραμματισμού που αντιμετωπίζουν κάποιες μόνο ειδικές περιπτώσεις ψευδοκόμπων, τις οποίες είναι ίσως πιθανό να συναντήσουμε στην πράξη. Έτσι, μια από τις πρώτες υλοποιήσεις αποτελεί ο αλγόριθμος **PKNOTS** http://selab.janelia.org/software/pknots/pknots.tar.gz (Rivas & Eddy, 1999). Το **CYLOFOLD** είναι ένας άλλος πιο σύγχρονος τέτοιος αλγόριθμος http://cylofold.abcc.ncifcrf.gov/ (Bindewald, Kluth, & Shapiro, 2010), όπως επίσης και το **KineFOLD** http://kinefold.curie.fr/cgi-bin/form.pl (Isambert, 2009), αλλά και το **IPknot** https://github.com/satoken/ipknot, (Sato, Kato, Hamada, Akutsu, & Asai, 2011). Τέλος, το **SimulFold**, http://www.cs.ubc.ca/~irmtraud/simulfold/, επιτυγχάνει κάτι παρόμοιο αλλά χρησιμοποιεί επιπλέον και πολλαπλές στοιχίσεις (Meyer & Miklós, 2007).



Εικόνα 10.9: *Ένα τυπικό παράδειγμα ψευδοκόμπου (https://en.wikipedia.org/wiki/Pseudoknot).*

Εκτός από τις διασταυρούμενες παλίνδρομες αλληλουχίες υπάρχουν και άλλες περιπτώσεις στις οποίες οι γραμματικές χωρίς συμφραζόμενα δεν επαρκούν. Ένα τέτοιο παράδειγμα είναι οι επαναληπτικές αλληλουχίες, όπως η *xxxyyyzzz*, η οποία προκύπτει από την επανάληψη *x*, *y* και *z* σε ίσους αριθμούς. Τέτοιες αλληλεπιδράσεις διασταυρώνονται αναγκαστικά και για να αποτυπωθούν τέτοιοι κανόνες, χρειάζεται μια γραμματική με συμφραζόμενα (context-sensitive grammar) ικανή να εκφράσει αυτή τη γλώσσα η οποία ονομάζεται αντιγραφική γλώσσα (copy language). Οι γλώσσες αυτές, και οι αντίστοιχες γραμματικές που τις περιγράφουν, έχουν μεγάλο θεωρητικό ενδιαφέρον κυρίως λόγω της μεγάλης τους περιγραφικής δύναμης. Παρ' όλα αυτά, οι υπερβολικές υπολογιστικές απαιτήσεις τέτοιων εφαρμογών έχουν μέχρι στιγμής λειτουργήσει αποτρεπτικά στην εφαρμογή τους σε βιολογικά προβλήματα.

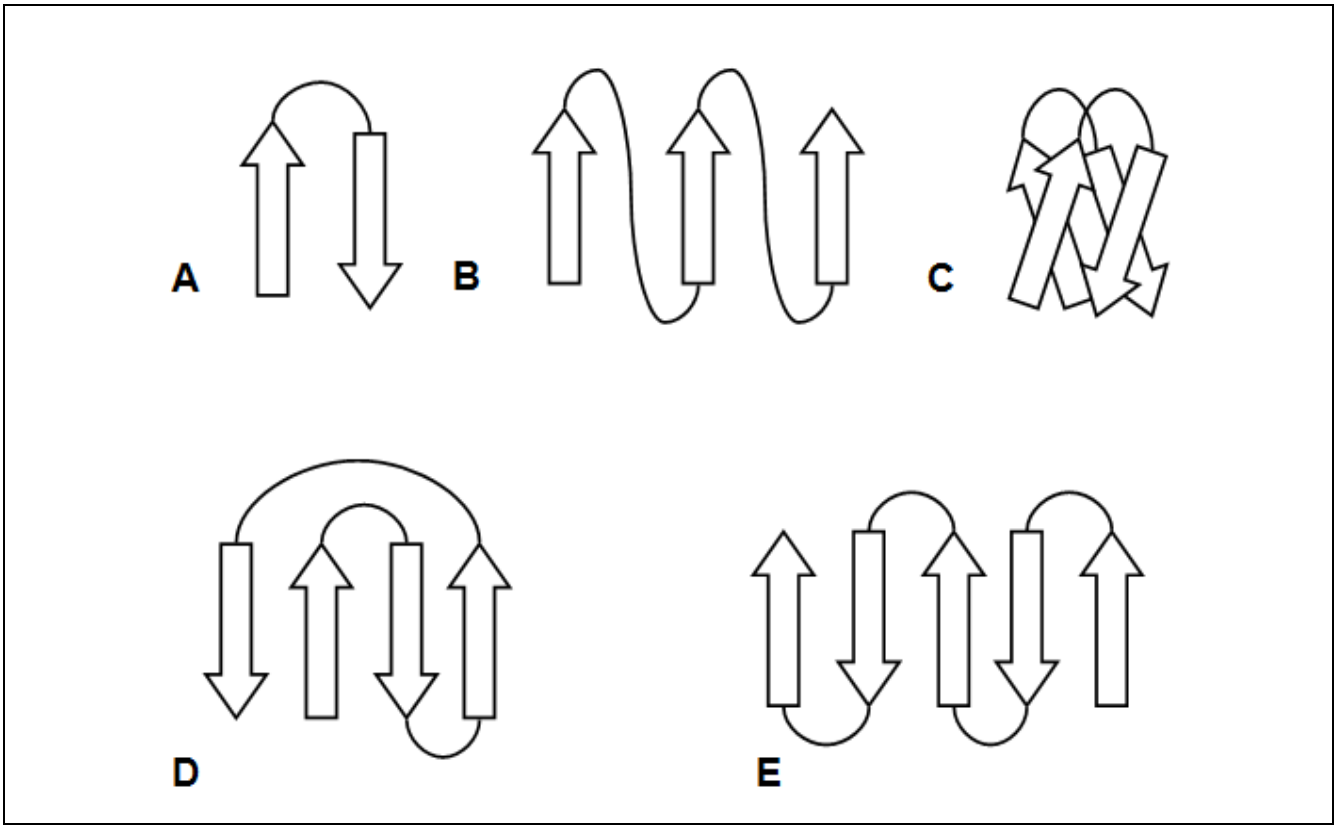
10.4. Εφαρμογές στην περίπτωση των πρωτεϊνών

Μέχρι τώρα ασχοληθήκαμε αποκλειστικά με την περίπτωση των RNA, τα οποία προσέφεραν το κλασικό παράδειγμα για την εφαρμογή των γραμματικών χωρίς συμφραζόμενα (παλίνδρομες γλώσσες και μακρινές αλληλεπιδράσεις). Παρ’ όλα αυτά, μακρινές αλληλεπιδράσεις και μάλιστα ιδιαίτερα πολύπλοκες, υπάρχουν και στην περίπτωση των πρωτεϊνών και μάλιστα είναι υπεύθυνες σε μεγάλο βαθμό για την πολυπλοκότητα των τρισδιάστατων δομών τους.

Η περίπτωση των πρωτεϊνών είναι πιο περίπλοκη, για μια σειρά από λόγους:

- Οι πρωτεΐνες έχουν μεγαλύτερο αλφάβητο (20 αμινοξέα αντί για 4 νουκλεοτίδια)
- Οι αλληλεπιδράσεις που σταθεροποιούν τη δομή είναι διαφόρων ειδών (δεσμοί υδρογόνου, υδρόφοβες αλληλεπιδράσεις, δεσμοί άλατος, αλληλεπιδράσεις Van der Waals)
- Δεν υπάρχει ξεκάθαρος κανόνας για τη συμπληρωματικότητα των αμινοξέων που συμμετέχουν σε αυτές, αν και φυσικά υπάρχουν προτιμήσεις

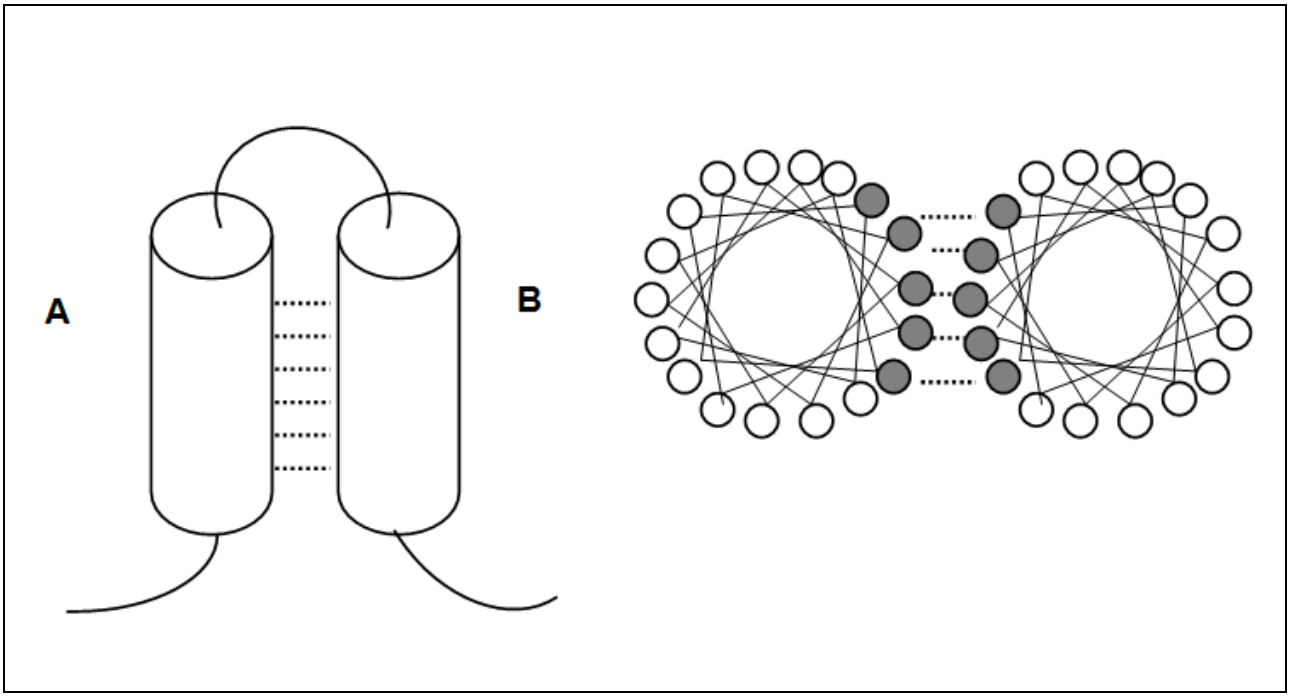
Ένα κλασικό παράδειγμα που προσεγγίζει όσο περισσότερο γίνεται την περίπτωση των αλληλεπιδράσεων των RNA, είναι η β-πτυχωτή επιφάνεια (Εικόνα 10.11). Υπάρχουν πολλές περιπτώσεις β-πτυχωτών επιφανειών, από την απλή φουρκέτα και τις παράλληλες και αντιπαράλληλες β-πτυχωτές επιφάνειες, μέχρι πιο σύνθετες δομές όπως το Greek-key motif αλλά και συνδυασμοί τέτοιων δομών για να δώσουν υπερ-δευτεροταγείς δομές, όπως το β-σάντουιτς, η β-προπέλα, το β-βαρέλι και η β-έλικα. Σε όλες τις περιπτώσεις, το χαρακτηριστικό γνώρισμα είναι ο σχηματισμός δεσμών υδρογόνου μεταξύ των N-H και C=O των πεπτιδικών δεσμών των αμινοξέων που βρίσκονται «απέναντι» στην αλυσίδα. Παρ’ όλα αυτά, υπάρχει η βασική διαφορά ότι το ζευγάρωμα αυτό δεν είναι αποκλειστικό, αν και φυσικά υπάρχουν προτιμήσεις στα ζευγάρια αμινοξέων. Επίσης, το ζευγάρωμα δεν είναι αποκλειστικό και με την έννοια ότι το ίδιο αμινοξύ εμπλέκεται σε ένα δεσμό υδρογόνου προς τη μία κατεύθυνση (με την ομάδα N-H), αλλά και προς την άλλη (με την ομάδα C=O).



Εικόνα 10.10: *Παραδείγματα δομών πρωτεϊνών τα οποία θα μπορούσαν να μοντελοποιηθούν με γραμματική χωρίς συμφραζόμενα.*

Όλα τα παραπάνω, καθιστούν την απλή εφαρμογή των μεθόδων που περιγράψαμε ήδη, ιδιαίτερα προβληματική. Παρ' όλα αυτά, η μελέτη τέτοιων περιπτώσεων είναι ιδιαίτερα σημαντική καθώς πιθανώς να ανοίξει το δρόμο προς την πρόγνωση της δομής πρωτεϊνών η οποία να προβλέπει και τις μακρινές αλληλεπιδράσεις, γιατί όπως είδαμε σε προηγούμενα κεφάλαια οι μέθοδοι πρόγνωσης έχουν ένα ανώτατο όριο επιτυχίας, καθώς αδυνατούν να λάβουν υπόψη τους τις μακρινές αλληλεπιδράσεις. Η πρώτη προσπάθεια εφαρμογής γραμματικών χωρίς συμφραζόμενα στην περίπτωση της πρόγνωσης δομής των πρωτεϊνών έγινε το 1994 από τους Mamitsuka και Abe (Mamitsuka & Abe, 1994). Οι συγγραφείς, επιχείρησαν να μοντελοποιήσουν τις β-πτυχωτές επιφάνειες, με όλους τους περιορισμούς που αναφέραμε παραπάνω, και κατέληξαν στη χρήση μιας ειδικής κατηγορίας γραμματικών, γνωστών και ως *Stochastic Ranked Node Rewriting Grammars* (SRNRG). Με τις γραμματικές αυτές και μια σειρά από τροποποιήσεις (μείωση αλφαβήτου αμινοξέων, χρήση μιας πιο γρήγορης εκδοχής του αλγορίθμου inside-outside, αλλά και παραλληλοποίηση των υπολογισμών), κατάφεραν να προβλέψουν με επιτυχία τις β-πτυχωτές επιφάνειες σε ένα σύνολο δεδομένων από τη βάση HSSP με λιγότερο από 25% ομοιότητα με το σύνολο εκπαίδευσης. Τα αποτελέσματα αυτά ήταν σημαντικά, γιατί όχι μόνο προβλέφθηκαν οι θέσεις των β-κλώνων αλλά το ίδιο έγινε και για τους δεσμούς υδρογόνου που σταθεροποιούν την επιφάνεια.

Βέβαια, οι υπολογιστικές απαιτήσεις τέτοιων εγχειρημάτων καθυστέρησαν αρκετά την εφαρμογή και εξάπλωση τέτοιων μεθόδων για περίπου μια δεκαετία. Το 2006, ο Searls με τους συνεργάτες του παρουσίασαν ένα γενικό πλαίσιο για την εφαρμογή γραμματικών κανόνων στην περιγραφή βιομοριακών αλληλουχιών, και ειδικά, στην περιγραφή των πρωτεϊνών (Chiang, Joshi, & Searls, 2006). Το 2005 ο Waldispuhl παρουσίασε μια ενδιαφέρουσα εφαρμογή των γραμματικών στην πρόγνωση των διαμεμβρανικών α-ελίκων (Waldispühl & Steyaert, 2005). Χρησιμοποίησε το λεγόμενο *multi-tape S-attributed grammar* το οποίο είναι για τις γραμματικές το αντίστοιχο του class HMM, καθώς αναθέτει στα μη-τερματικά σύμβολα μια «ιδιότητα», η οποία στη συγκεκριμένη περίπτωση συμβόλιζε την αλληλεπίδραση με τις γειτονικές έλικες. Με τη μέθοδο αυτή (**TMMTSAG**), πέτυχαν συγκρίσιμα αποτελέσματα στην πρόβλεψη των διαμεμβρανικών α-ελίκων, σε σχέση με τις υπάρχουσες μεθόδους, αλλά το σημαντικότερο ήταν ότι προέβλεψαν και την αλληλεπίδραση των αμινοξέων με τις άλλες έλικες (ή την έκθεση προς τα λιπίδια της μεμβράνης). Δυστυχώς η μέθοδος αυτή δεν είναι δημόσια διαθέσιμη.



Εικόνα 10.11: *Παράδειγμα αλληλεπίδρασης μεταξύ δύο α-ελίκων (διαμεμβρανικές ή μη). Α. Η αναπαράσταση των δύο ελίκων με το κλασικό μοντέλο. Β. Αναπαράσταση με το helical wheel plot, το οποίο δείχνει τα αμινοξέα από τον κάθετο άξονα της έλικας. Με γκρι φαίνονται τα αμινοξέα τα οποία βρίσκονται σε επαφή μεταξύ τους, ενώ με άσπρο τα αμινοξέα που βρίσκονται σε επαφή με τον περιβάλλοντα χώρο. Λόγω της περιοδικότητας της έλικας, ένα κάθε τέσσερα ή πέντε αμινοξέα αναμένουμε να είναι «γκρι», αλλά στο χώρο αυτά συσσωρεύονται σε μια επιφάνεια επαφής.*

Η ίδια ομάδα, επιχείρησε να εφαρμόσει τις ίδιες μεθοδολογίες και στην περίπτωση των διαμεμβρανικών β-βαρελιών. Έτσι, δημιουργήθηκε η μέθοδος **transFold** (http://bioinformatics.bc.edu/clotelab/transFold) η οποία προβλέπει με αρκετά ικανοποιητικό τρόπο, τουλάχιστον σε σύγκριση με τους υπόλοιπους αλγόριθμους του είδους, τους διαμεμβρανικούς β-κλώνους αλλά και το δίκτυο των δεσμών υδρογόνου που σταθεροποιούν το β-βαρέλι (Waldispuhl, Berger, Clote, & Steyaert, 2006). Ένα μειονέκτημα της μεθόδου είναι το γεγονός ότι είναι αργή, αλλά και ότι απαιτεί διάφορες παραμέτρους από τον χρήστη (τι είδους βαρέλι περιμένουμε να βρούμε κ.ο.κ.). Μια επέκταση της μεθοδολογίας αυτής, παράλληλα με χρήση ενός επιπλέον βήματος για την ελαχιστοποίηση ενέργειας, αποτελεί η μέθοδος **Partifold** η οποία είναι διαθέσιμη στη διεύθυνση http://partiFold.csail.mit.edu/ (Waldispuhl, O'Donnell, Devadas, Clote, & Berger, 2008).

Τέλος, αξίζει να αναφερθούν και οι εργασίες των Dyrka και Nebel, οι οποίοι ανέπτυξαν ένα ολόκληρο πλαίσιο για την εφαρμογή γραμματικών χωρίς συμφραζόμενα σε παρόμοια προβλήματα πρωτεϊνών, ακολουθώντας μια διαφορετική στρατηγική. Το βασικό χαρακτηριστικό της μεθόδου αυτής, ήταν η χρήση ενός γενετικού αλγόριθμου για να μπορέσει να εξάγει τους κανόνες της γραμματικής, αλλά και η χρήση μιας διαφορετικής αναπαράστασης βασισμένης στις ιδιότητες των αμινοξέων προκειμένου να μειωθεί το αλφάβητο (Dyrka & Nebel, 2009). Η μεθοδολογία αυτή, εφαρμόστηκε σε μια σειρά από εργασίες, τόσο για την πρόγνωση των αλληλεπιδράσεων των διαμεμβρανικών ελίκων (Dyrka, Nebel, & Kotulska, 2013), όσο και για την πρόγνωση θέσεων πρόσδεσης άλλων μορίων πάνω στις πρωτεΐνες (Dyrka & Nebel, 2007). Όλα τα παραπάνω, δείχνουν ότι οι μεθοδολογίες αυτές αν και δεν έχουν εφαρμοστεί ιδιαίτερα λόγω των εγγενών δυσκολιών στις πρωτεΐνες, εν τούτοις είναι πολύ υποσχόμενες και καθώς η υπολογιστική ισχύς αυξάνεται, αλλά και η έρευνα στους αλγόριθμους συνεχίζεται, αναμένεται στο μέλλον να παίξουν σημαντικό ρόλο στην επίλυση τέτοιων προβλημάτων.

Βιβλιογραφία

Baker, J. K. (1979). Trainable grammars for speech recognition. The Journal of the Acoustical Society of America, 65(S1), S132-S132.

Bindewald, E., Kluth, T., & Shapiro, B. A. (2010). CyloFold: secondary structure prediction including pseudoknots. Nucleic Acids Research, 38(suppl 2), W368-W372.

Chiang, D., Joshi, A. K., & Searls, D. B. (2006). Grammatical representations of macromolecular structure. Journal of computational biology, 13(5), 1077-1100.

Chomsky, N. (1956). Three models for the description of language. IRE Transactions on Information Theory, 2(3), 113-124.

Chomsky, N. (1959). On Certain Formal Properties of Grammars. Information and Control 2(2), 137-167.

Do, C. B., Woods, D. A., & Batzoglou, S. (2006). CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics, 22(14), e90-e98.

Dowell, R. D., & Eddy, S. R. (2004). Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. BMC Bioinformatics, 5(1), 71.

Durbin, R., Eddy, S. R., Krogh, A., & Mithison, G. (1998). Biological sequence analysis, probabilistic models of proteins and nucleic acids: Cambridge University Press.

Dyrka, W., & Nebel, J.-C. (2007). A probabilistic context-free grammar for the detection of binding sites from a protein sequence. Bmc Systems Biology, 1(Suppl 1), P78.

Dyrka, W., & Nebel, J.-C. (2009). A stochastic context free grammar based framework for analysis of protein sequences. BMC Bioinformatics, 10(1), 323.

Dyrka, W., Nebel, J.-C., & Kotulska, M. (2013). Probabilistic grammatical model for helix-helix contact site classification. Algorithms for Molecular Biology, 8(1), 31.

Eddy, S. R. (2004). How do RNA folding algorithms work? Nature biotechnology, 22(11), 1457-1458.

Eddy, S. R., & Durbin, R. (1994). RNA sequence analysis using covariance models. Nucleic Acids Res, 22(11), 2079-2088.

Gardner, P. P., Daub, J., Tate, J., Moore, B. L., Osuch, I. H., Griffiths-Jones, S., . . . Bateman, A. (2011). Rfam: Wikipedia, clans and the "decimal" release. Nucleic Acids Res, 39(Database issue), D141-145.

Isambert, H. (2009). The jerky and knotty dynamics of RNA. Methods, 49(2), 189-196.

Ito, K., Murakami, R., Mochizuki, M., Qi, H., Shimizu, Y., Miura, K.-i., . . . Uchiumi, T. (2012). Structural basis for the substrate recognition and catalysis of peptidyl-tRNA hydrolase. Nucleic Acids Research, gks790.

Knudsen, B., & Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. Nucleic Acids Research, 31(13), 3423-3428.

Lange, M., & Leiß, H. (2009). To CNF or not to CNF? An efficient yet presentable version of the CYK algorithm. Informatica Didactica, 8, 2008-2010.

Lorenz, R., Bernhart, S. H., Zu Siederdissen, C. H., Tafer, H., Flamm, C., Stadler, P. F., & Hofacker, I. L. (2011). ViennaRNA Package 2.0. Algorithms for Molecular Biology, 6(1), 26.

Lyngsø, R. B., & Pedersen, C. N. (2000). RNA pseudoknot prediction in energy-based models. Journal of computational biology, 7(3-4), 409-427.

Mamitsuka, H., & Abe, N. (1994). Predicting location and structure of beta-sheet regions using stochastic tree grammars. Proc Int Conf Intell Syst Mol Biol, 2, 276-284.

Meyer, I. M., & Miklós, I. (2007). SimulFold: simultaneously inferring RNA structures including pseudoknots, alignments, and trees using a Bayesian MCMC framework.

Nawrocki, E. P., Kolbe, D. L., & Eddy, S. R. (2009). Infernal 1.0: inference of RNA alignments. Bioinformatics, 25(10), 1335-1337.

Nussinov, R., Pieczenik, G., Griggs, J. R., & Kleitman, D. J. (1978). Algorithms for Loop Matchings. SIAM Journal on Applied Mathematics 35(1), 68-82.

Pedersen, J. S., Bejerano, G., Siepel, A., Rosenbloom, K., Lindblad-Toh, K., Lander, E. S., . . . Haussler, D. (2006). Identification and classification of conserved RNA secondary structures in the human genome. PLoS Comput Biol, 2(4), e33.

Rivas, E., & Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. Journal of molecular biology, 285(5), 2053-2068.

Rivas, E., Lang, R., & Eddy, S. R. (2012). A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. RNA, 18(2), 193-212.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjolander, K., Underwood, R. C., & Haussler, D. (1994). Stochastic context-free grammars for tRNA modeling. Nucleic Acids Res, 22(23), 5112-5120.

Sato, K., Kato, Y., Hamada, M., Akutsu, T., & Asai, K. (2011). IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. Bioinformatics, 27(13), i85-i93.

Searls, D. B. (2002). The language of genes. Nature, 420(6912), 211-217.

Waldispuhl, J., Berger, B., Clote, P., & Steyaert, J. M. (2006). transFold: a web server for predicting the structure and residue contacts of transmembrane beta-barrels. Nucleic Acids Res, 34(Web Server issue), W189-193.

Waldispuhl, J., O'Donnell, C. W., Devadas, S., Clote, P., & Berger, B. (2008). Modeling ensembles of transmembrane beta-barrel proteins. Proteins, 71(3), 1097-1112.

Washietl, S., Hofacker, I. L., & Stadler, P. F. (2005). Fast and reliable prediction of noncoding RNAs. Proceedings of the National Academy of Sciences of the United States of America, 102(7), 2454-2459.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n 3. Information and Control, 10(2), 189-208.

Zuker, M., & Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res, 9(1), 133-148.