Le langage des molécules du vivant

Jacques Nicolas, Catherine Belleannée, François Coste

Revue Tangente
Hors-série 52 Bibliothèque "Mathématiques et Informatique"

Le langage des molécules du vivant

Auteurs : Jacques Nicolas, Catherine Belleannée et François Coste
IRISA / INRIA Centre de Rennes Bretagne Atlantique

Voyage au cœur des cellules

Depuis une quinzaine d'années, la biologie dispose de moyens d'observation sans précédent pour accéder au niveau moléculaire le plus intime à l'ensemble des informations présentes dans une cellule. La question centrale de la biologie est de passer de cette information de bas niveau à la compréhension des fonctions essentielles du vivant : la reproduction, l'autoréparation, l'organisation en structures emboîtées à l'aide de membranes isolant les systèmes cruciaux, et enfin l'évolution et l'adaptation au milieu de vie qui permettent aux organismes d'affronter des conditions d'environnement très variées.

Mais de quelles informations dispose-t-on exactement ?

Le déferlement des données génomiques concerne non seulement le code des chromosomes humains -obtenu en 2003-, mais également celui d'un nombre croissant d'organismes, des plus petits, les virus et les bactéries, aux plus gros comme l'éléphant africain. A la base il y a donc *la cellule*, la brique de base de tout organisme vivant, et au cœur de la cellule, ces longues molécules d'ADN qui forment les *génomes* et qui recèlent toutes les données nécessaires à la maintenance et à la reproduction de ces cellules. En termes mathématiques, un génome peut être vu de manière simplifiée comme une (grande) séquence écrite sur un alphabet à quatre lettres, *les bases a, c, g* et *t*. C'est un peu plus compliqué, la séquence est orientée (le texte a un sens de lecture) et c'est un agent double : l'ADN a deux brins enlacés, qui sont codés de manière totalement complémentaire, c'est-à-dire en sens opposé et en associant les lettres qui se font face selon un schéma rigide de paires *a-t* ou *g-c*.

Nous savons maintenant que l'information nécessaire à la compréhension des comportements vivants est loin de s'arrêter aux chromosomes. La molécule d'ADN est une sorte de conservatoire, elle code et retient l'information héréditaire comme un notaire veillant sur le patrimoine de ses clients, mais il faut bien avouer qu'elle n'est pas très active. Une première molécule semblable à l'ADN mais beaucoup plus dynamique est l'ARN, probablement apparue très tôt dans la soupe primordiale de molécules ayant conduit aux formes de vie terrestre. Les molécules d'ARN, de différents types, sont les agents de traduction du message génétique en mesures concrètes et adaptées à l'intérieur de la cellule. Ainsi, les segments d'ADN correspondant à une entité appelée *gène* pourront donner naissance à différentes variantes en ARN (appelées *transcrits*) suivant le contexte dans lequel se trouve la cellule. L'ARN participe également à la formation des machines moléculaires qui vont traduire puis réguler l'expression des gènes pour former des molécules actives. Formellement, l'ARN forme aussi un texte orienté à quatre lettres, *a, c, g* et *u*. Simple brin cette fois et plus court, il peut se replier dans l'espace en s'appariant sur lui-même (paires *a-u* ou *g-c*) et prendre des formes variées caractéristiques de la fonction de la molécule. On sait maintenant observer finement la présence d'ARN dans les cellules, en terme non seulement du texte mais des quantités produites. La structure spatiale est difficilement accessible par des méthodes expérimentales indirectes et on s'appuie en général sur la modélisation mathématique pour la découvrir.

Il nous faut évoquer enfin le troisième grand type de (macro-)molécule des cellules, les *protéines*, qui sont les agents à tout faire de la cellule : elles ont un rôle structurel, elles s'occupent du transport des molécules, de la signalisation des évènements extérieurs, de la mobilité, elles catalysent les réactions chimiques et régulent l'expression des gènes. Les protéines sont produites par traduction d'un type d'ARN dit messager. Les textes en résultant sont codés sur un alphabet, *les acides aminés* (une vingtaine de lettres) et ne dépassent pas quelques milliers de caractères. Les propriétés physico-chimiques de ces molécules sont beaucoup plus variées que celles des molécules d'ADN et d'ARN, de même que les structures qu'elles adoptent spontanément dans l'espace. On sait observer l'ensemble des protéines présentes dans un échantillon. Connaître leur structure spatiale est un problème à la fois difficile expérimentalement et fondamental en pratique car la fonction des protéines est intimement liée à cette structure. Comme pour l'ARN, la modélisation mathématique et informatique se révèle donc indispensable à leur étude, en particulier pour prédire la structure à partir du texte.

Des textes naturels et troublants…

Pour un informaticien, observer la complexité des mécanismes du vivant est une source infinie d'étonnement et d'inspiration. Car il ne s'agit pas uniquement de physique et de chimie. Les différents textes à l'intérieur des cellules, qui sont le fruit d'un longue maturation tout au long de l'évolution, ont un impact majeur sur le comportement cellulaire. Dans des cas extrêmes, le changement d'une seule base d'ADN conduit à des états pathologiques, comme c'est le cas pour la première maladie génétique de France, la drépanocytose, où la mutation d'une base à l'intérieur d'un gène entraîne un changement d'acide aminé de la protéine associée et mène à une déformation importante des globules rouges.

Si on regarde ces textes d'un peu plus près, on s'aperçoit qu'ils sont tout sauf aléatoires. La première impression qui frappe est l'abondance de répétitions plus ou moins exactes qui peuplent les séquences biologiques. On sait maintenant qu'il existe des mécanismes dans les cellules commandés par des protéines propres - les transposases- ou importées de virus –les intégrases- qui permettent de copier certaines portions de génome, *les transposons*, à un autre endroit dans le génome. D'un point de vue informatique, il s'agit ni plus ni moins d'une version du copier/coller ou du couper/coller que vous utilisez sans doute régulièrement sur votre ordinateur ou votre mobile, à ceci près que le texte copié contient aussi le programme de copie ! C'est un mécanisme très efficace qui provoque la croissance des génomes et apporte un éclairage sur leur étonnante plasticité et leur capacité à évoluer et à se diversifier au-delà de ce que permettraient les seuls accidents de transmission du patrimoine génétique (les mutations) et les recombinaisons entre individus lors de la reproduction. Le record de croissance, $1{,}5 \times 10^{11}$ bases d'ADN, est actuellement détenu par une plante, *Paris japonica*, qui pousse sur les montagnes de l'île de Honshu (le génome humain n'a « que » 3×10^{9} bases).

D'une manière générale, il existe d'autres mots que l'on retrouve de façon assez systématique dans ces textes et qui les charpentent, et qui sont presque toujours associés à des machines plus ou moins complexes (à base d'ARN et de protéines), qui vont permettre la lecture et la transformation des textes, un peu à la manière des têtes de lecture/écriture de nos enregistreurs électroniques. On pourrait multiplier les exemples. Ainsi, les fameux gènes, unités génomiques qui vont finir par être traduites en protéines, sont découpés chez les organismes supérieurs en tronçons appelés exons et introns qui s'alternent le long de la molécule. Le texte utile pour la protéine sera constitué d'une sélection d'exons qui seront mis bout à bout. Sans comprendre en détails comment la machine associée, le *splicéosome*, effectue cette sélection, on connait déjà des mots qui se retrouvent majoritairement aux jonctions entre les tronçons : *gu* au début et *agg* à la fin des introns par exemple. Pour le passage du texte en ARN au texte en acides aminés, la machine s'appelle *ribosome* et le début du texte à traduire commence par

aug et parfois *gug* ou *uug* et finit invariablement par *uag*, *uga* ou *uaa*. Systématiquement, les triplets de lettres d'ARN vont être transformés en une lettre d'acide aminé. On observe là des phénomènes très différents de ce qui se passe en chimie classique : de véritable codes sont interprétés de manière rigoureuse dans la cellule. Autre exemple de machine en action sur les cellules germinales ou les cellules cancéreuses, les *télomérases*. Il s'agit d'un assemblage de protéines et d'ARN chargé d'ajouter une séquence répétée spécifique à l'extrémité des chromosomes, pour éviter de perdre des gènes lors de leur réplication pour produire de nouvelles cellules. Il est certain qu'il reste encore de telles machines à découvrir au cœur des cellules. Parmi les découvertes dont on a récemment compris l'importance, nous évoquerons juste les structures de *CRISPR*, au nom improbable mais dont l'existence est encore plus surprenante : on trouve dans bon nombre de bactéries des séries de mots répétés d'une trentaine de bases qui forment une ossature dans laquelle vont s'intercaler d'autres mots très différents, qui semblent issus d'une langue étrangère à la bactérie. En fait, tout comme nous, les bactéries sont sujettes à des attaques de virus et cette structure se comporte comme un ensemble de cases mémoires, qui retiennent des morceaux de séquences de virus rencontrés afin de les reconnaître lors de futures attaques : ce n'est ni plus ni moins qu'un système immunitaire auquel sont associés un ensemble de mots et une machinerie spécifique de gènes associés.

De façon claire et remarquable, les mots ainsi que leur assemblage et leur interprétation par des machines spécialisées jouent un rôle prédominant dans les cellules vivantes. Au-delà des phénomènes de thermodynamique, d'électrostatique ou autres lois à caractère continu, il existe des représentations et des comportements de nature symbolique qui contrôlent le devenir des cellules.

Langages, structures et interactions

Des mots qui s'enchaînent de façon contrôlée, y aurait-il donc un langage, voire des langages à l'œuvre dans les organismes ? Précisons les termes employés : il y a généralement deux niveaux d'écriture dans une langue, le niveau lexical qui est celui des mots du dictionnaire, et le niveau syntaxique, qui est celui des enchaînements possibles au sein de phrases. L'ensemble des mots est généralement fini et on parlera ici de *vocabulaire* pour indiquer l'ensemble des symboles qu'on souhaite manipuler. Le vocabulaire peut être constitué de simples caractères (des idéogrammes comme les hiéroglyphes, ou des acides aminés pour les protéines) ou de succession de caractères (un mot en français, une balise HTML), peu importe, on le suppose figé. Une façon particulièrement simple de décrire un langage est d'établir la liste des phrases qu'il contient, c'est-à-dire des combinaisons permises d'éléments du vocabulaire. Mais contrairement aux vocabulaires, les langages peuvent la plupart du temps être considérés comme infinis. On ne peut donc pas se contenter d'énumérer leurs éléments. L'informatique a été très tôt confrontée à ce problème car il y a un lien intime entre langages et calcul. Après tout, la logique mathématique n'est rien d'autre que l'étude des mathématiques en tant que langage et un ordinateur est une machine de manipulation de langages. La *théorie des langages* a pour but de décrire formellement la notion de langage. En espérant ne pas réveiller des souvenirs douloureux pour certains, il nous faut maintenant introduire un outil commun des linguistes et des informaticiens pour générer avec un système formel fini un nombre potentiellement non borné de phrases : les grammaires. Formellement, une grammaire est un ensemble de règles de récriture qui repose à la fois sur l'utilisation d'un ensemble de symboles spécifiques appelé non-terminaux et d'un vocabulaire dont les éléments sont dits terminaux. Pour générer toutes les phrases du langage, il suffit de transformer itérativement une chaîne réduite initialement à un symbole fixé, l'axiome, et d'appliquer de toutes les manières possibles les règles en changeant une occurrence d'une partie gauche de la règle dans cette chaîne par la partie droite de la règle.

Par exemple, en prenant S comme axiome, une grammaire possible de génération d'une séquence d'ARN bactérien codant pour une protéine est la suivante :

$\{1 : S \rightarrow a\ X_1,\ 2 : X_1 \rightarrow u\ X_2,\ 3 : X_2 \rightarrow g\ X_3\} \cup \{4 : X_3 \rightarrow \alpha\ X_4,\ 5 : X_7 \rightarrow \alpha\ X_4 \mid \alpha \in \{a,c,g\}\} \cup \{6 : X_4 \rightarrow \beta\ X_6,\ 7 : X_6 \rightarrow \beta\ X_7 \mid \beta \in \{a,c,g,u\}\} \cup \{8 : X_5 \rightarrow \gamma X_6,\ 9 : X_8 \rightarrow \gamma X_7 \mid \beta \in \{c,g,u\}\} \cup \{10 : X_3 \rightarrow uX_5,\ 11 : X_7 \rightarrow u\ X_5,\ 12 : X_5 \rightarrow a\ X_8,\ 13 : X_8 \rightarrow a\}$

Elle pourra par exemple générer la séquence *augccguaa* en utilisant successivement les règles 1, 2 et 3 (*aug*), puis 4, 6 et 7 (*ccg*), et enfin 11, 12 et 13 (*uaa*). Observez que la structure des règles est très régulière : elles sont toutes de la forme « un non terminal se récrit en un symbole terminal suivi éventuellement d'un non terminal ». On nomme réguliers les langages générés à partir d'une grammaire avec cette forme de règles. On peut montrer que ces langages ont une foule de bonnes propriétés qui en font un outil précieux en informatique (par exemple pour le système Unix ou les traitements de texte). Ainsi, savoir si une phrase appartient à un langage régulier demande un nombre d'opérations proportionnel à la taille de la phrase. De plus ils forment une classe stable au sens où l'intersection, l'union, le complémentaire, la différence ou l'application d'un homomorphisme sur un langage régulier continuent à donner un langage régulier.

De façon complètement équivalente à la représentation grammaticale, on peut décrire les langages à l'aide de machines : c'est un modèle plus proche de ce qui se passe en biologie où nous avons observé de nombreuses machines en œuvre ainsi qu'en informatique où on définit plusieurs types de machines abstraites en fonction du type de langages, la plus générale, la *machine de Türing*, étant le fondement de nos ordinateurs. Pour reconnaître un langage régulier, on utilise ainsi ce qu'on appelle des *automates d'états finis*. La machine part d'un état initial et lit les symboles de la phrase de gauche à droite. En utilisant une fonction de transition fixée qui associe à chaque état et chaque symbole lu un nouvel état, la machine progresse tant que c'est possible d'états en états. La phrase est reconnue si la machine termine dans un état final. On représente graphiquement les états par des cercles, un état final par un double cercle et une transition en lisant un symbole par une flèche depuis l'état de départ jusqu'à l'état d'arrivée surmontée du symbole. Nous effleurons juste en passant la notion de probabilité qu'on peut introduire dans les langages et leur représentations : rien n'empêche de considérer un langage comme une distribution de probabilités sur l'ensemble des enchaînements possibles. D'un point de vue grammaire ou machine, ceci suppose d'associer des probabilités aux règles ou aux transitions. Nous nous contentons ici de considérer que toutes les phrases ont la probabilité 0 ou 1. En pratique, il peut exister en biologie différentes alternatives d'analyse (on parle d'ambiguïté) avec des probabilités différentes pour de mêmes phrases, comme par exemple dans le cas des télomérases qui oscillent entre 2 états stables.

Cependant la classe des langages réguliers reste trop limitée pour décrire ce qui se passe en biologie. Il n'est par exemple pas possible de décrire les structures en forme de tige qui naissent lors du repliement de l'ARN dans l'espace pour former des double brins, comme dans certaines structures de CRISPR (une structure qui ressemble fort aux palindromes du français). Pour cela, il faut pouvoir mettre en correspondance des symboles éventuellement distants. On étend donc les règles de grammaires régulières en permettant un assemblage quelconque de terminaux et de non terminaux dans la partie droite. Comme les non terminaux se récrivent directement où qu'ils soient, on parle de *grammaire hors contexte*. Ainsi, une grammaire hors contexte pour la reconnaissance de tiges-boucles dans l'ARN peut être décrit par deux ensembles de règles (l'un pour la tige, le deuxième pour la boucle) : $\{S \rightarrow a\ S\ u,\ S \rightarrow c\ S\ g,\ S \rightarrow g\ S\ c,\ S \rightarrow u\ S\ a\} \cup \{S \rightarrow \alpha X,\ X \rightarrow \alpha \mid a \in \{a,c,g,t\}\}$. De même, il faut améliorer la machine précédente en ajoutant une mémoire (supposée infinie) où on peut empiler des symboles spéciaux qui sont ensuite utilisés pour définir la fonction de transition. La

richesse de description a un coût : reconnaître une phrase d'un langage peut demander un nombre d'opérations de l'ordre de n^3 si n est la longueur de la phrase.

Notons ici que les langages permettent de structurer les textes de façon logique mais également spatiale dans le cas des macromolécules : à une correspondance entre symboles correspond vraiment une proximité physique et/ou une interaction chimique. Au bout du compte, un langage est porteur de sens et savoir le décrire donne des indications précieuses pour comprendre les phénomènes à l'œuvre.

Les langages de programmation et la description des pages web reposent sur ce type de grammaire, mais est-ce suffisant pour la biologie ? La réponse est clairement non. Les appariements qu'acceptent les grammaires hors contexte doivent être nécessairement emboîtés (par exemple si j'ouvre une parenthèse [voire ici une deuxième], je suis obligé de mettre le crochet fermant avant la parenthèse fermante). Or si on considère des structures d'ARN comportant des pseudo-nœuds comme dans les télomérases, ou des protéines comportant plusieurs acides aminés appelés cystéines qui ont tendance à s'assembler spontanément pour former des liaisons fortes, les appariements peuvent être dans un ordre quelconque par rapport à l'ordre dans la phrase. Il faut encore augmenter la complexité des langages et des grammaires nécessaires. Les *langages contextuels* sont engendrés par des grammaires étendues où le symbole non terminal de gauche et la partie droite de la règle peut être encadrée d'autant de symboles que nécessaire (il a un contexte d'application) du moment qu'on les retrouve à droite. Les langues naturelles comme les langages moléculaires du vivant se rangent dans cette catégorie, sans exiger cependant toute la puissance offerte par les grammaires contextuelles. Savoir découvrir et modéliser au juste niveau ces langages de façon à analyser automatiquement leurs phrases reste un passionnant challenge de recherche.

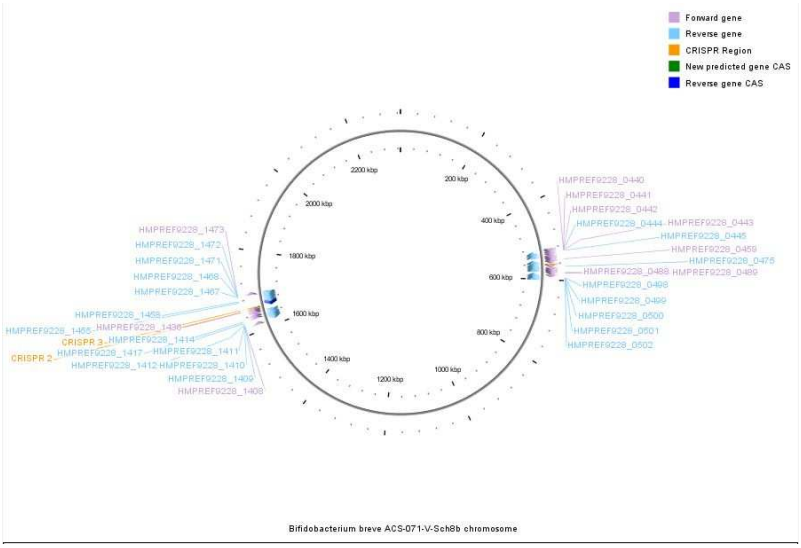
Encadré Focus : découvrir un langage, c'est automatisable ?

L'acquisition d'un langage, c'est un jeu d'enfant : l'essentiel est achevé avant l'âge de 3 ans pour la plupart. Si on arrive à apprendre n'importe quelle langue naturelle, ne peut-on arriver à apprendre cette autre langue naturelle qu'est le langage du génome ? On est cependant loin de comprendre les mécanismes internes qui permettent cette acquisition ni même de savoir comment nous représentons mentalement les langages et il nous faut inventer une approche rationnelle de l'apprentissage. Notons que la démarche n'est pas tout à fait celle d'un Champollion déchiffrant les hiéroglyphes : on ne dispose pas d'un langage de référence sur lequel s'appuyer pour comprendre un nouveau langage (en théorie des langages on parlerait de *transducteurs*), il s'agit bien d'apprendre un nouveau langage simplement à partir d'un échantillon de phrases.

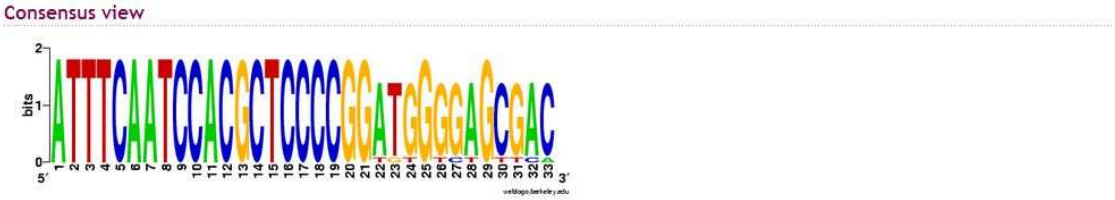
On sait actuellement assez bien modéliser l'apprentissage de langages réguliers. Une idée simple mais mathématiquement intéressante est de partir d'un langage qui reconnaît uniquement les phrases autorisées. C'est par exemple l'automate de gauche dans la figure. Ensuite, on peut généraliser le langage correspondant (c'est-à-dire accepter plus de phrases) en fusionnant 2 états quelconques (ils deviennent égaux et on conserve l'union des transitions auxquelles ils participent). La structure mathématique de l'ensemble des possibilités est ce que l'on appelle un treillis, c'est-à-dire un ensemble partiellement ordonné où tout couple d'éléments admet une borne supérieure et une borne inférieure. Si on considère l'ensemble de tous les états E, c'est en fait le treillis des partitions sur E, c'est-à-dire l'ensemble des façons de partitionner E en sous-ensembles distincts, ordonné par l'inclusion sur les partitions. Généraliser, c'est aller de la gauche vers la droite dans le treillis. Pour savoir jusqu'où le faire et éviter de reconnaître n'importe quel enchaînement (automate de droite sur la figure), on peut par exemple dialoguer avec l'utilisateur en proposant l'automate le plus général et en demandant une phrase impossible si le langage est jugé trop permissif, afin de filtrer progressivement la bonne solution.

Illustrations

Une structure de CRISPR dans une bactérie



Chromosome circulaire de *Bifidobacterium breve*, une bactérie lactique qui facilite la digestion chez les nourrissons. Emplacement des structures de CRISPR et les gènes de la machinerie associée.



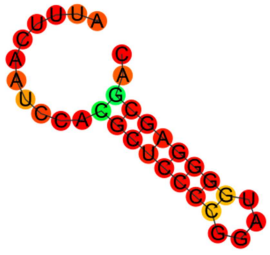
Crispr detail

Show repeat palindromic structure

Select all units | Select all spacers | Clear all | Download selected data | Extract flanking sequences | Extract CRISPR sequence

		BEGIN	END	SEQUENCE	SIZE
	unit 1	1654072	1654104	ATTTCAATCCACGCTCCCCGGATGGGGAGCGAC ...........(((((((((....)))))))))..	33
	spacer 1	1654105	1654140	GACCTCGCGGACCTCATCCTCGACACCATCAGCAGG	36
	unit 2	1654141	1654173	ATTTCAATCCACGCTCCCCGGATGGGGAGCGAC ...........(((((((((....)))))))))..	33
	spacer 2	1654174	1654206	GGTGGCGGGAAACATCATCGGTCGTGGACAAAT	33
	unit 3	1654207	1654239	ATTTCAATCCACGCTCCCCGGATGGGGAGCGAC ...........(((((((((....)))))))))..	33
	spacer 3	1654240	1654274	TGCGCCATATCCCCTTTGATCTGCCCGTCAAGGCG	35
	unit 4	1654275	1654307	ATTTCAATCCACGCTCCCCGGATGGGGAGCGAC ...........(((((((((....)))))))))..	33
	spacer 4	1654308	1654342	CGTTCGACTGCGCGGCGATGTGCGCCGTGTACAAG	35
	unit 5	1654343	1654375	ATTTCAATCCACGCTCCCCGGATGGGGAGCGAC ...........(((((((((....)))))))))..	33

Détail des répétitions (unit) d'un CRISPR, sa structure emboîtée, et les segments viraux (spacer).

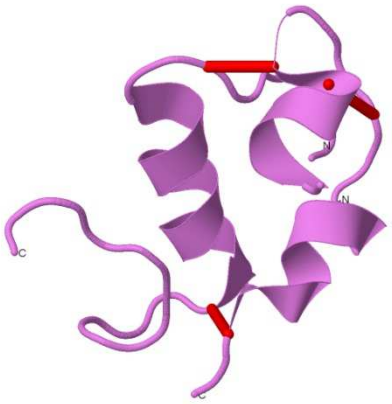


La structure en tige-boucle des répétitions du CRISPR précédent.

Source : Auteur (J.Nicolas), base de données http://crispi.genouest.org/

Motifs :

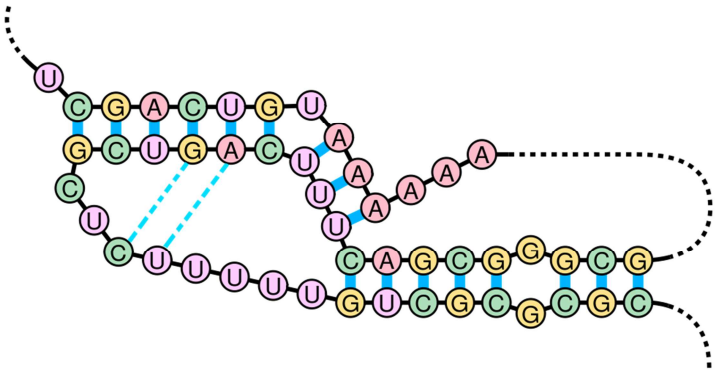
- **dans une protéine : insuline.**



Les protéines de la famille de l'insuline partagent toutes une structure spatiale commune de deux chaînes liées d'acides aminés. La première chaîne respecte un motif de séquence C_1-C_2-x(3)-C_3-[STDNEKPI]-x(3)-[LIVMFS]-x(3)-C_4, la deuxième un motif C_5-x(12)-C_6. de plus, les cystéines C_i forment des liaisons, les ponts disulfures (en rouge dans l'image) qui rigidifient la forme de la molécule dans l'espace : C_1 est reliée à C_3, C_2 est reliée à C_5, et C_3 est reliée à C_6.

Source : Auteur (J.Nicolas) de la base de données PDB

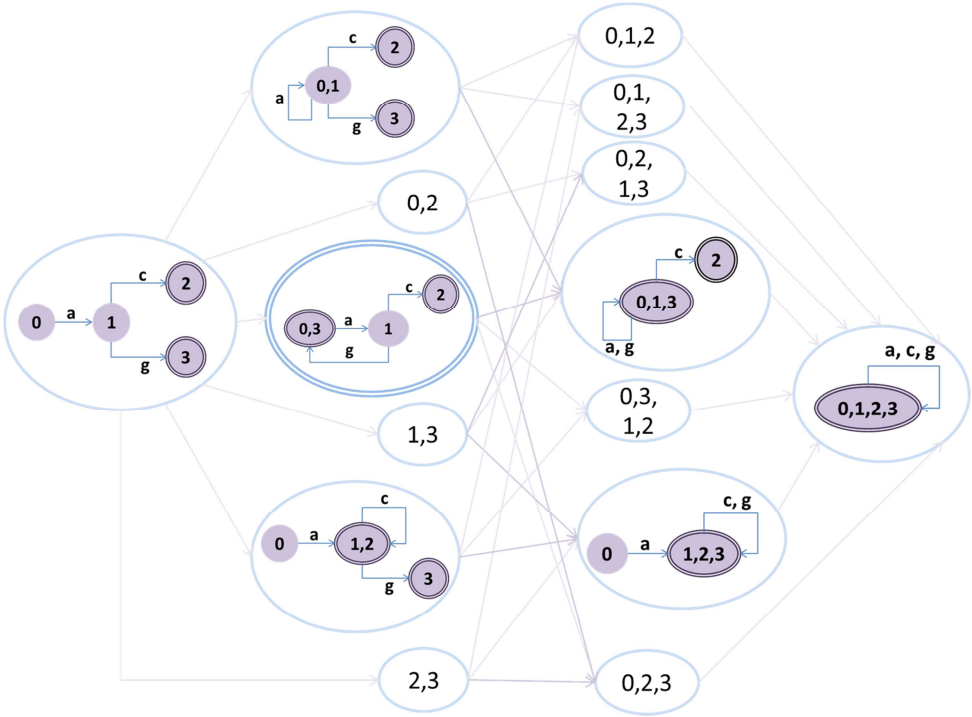
- **dans un ARN : télomérase.**



Une structure de pseudo-nœud dans la partie ARN de la télomérase humaine : cette structure permet de positionner une matrice qui va générer des répétitions TTAGGG à la fin des chromosomes et éviter le raccourcissement de leurs extrémités lors de leur réplication (mécanisme également utilisé dans les cancers). En pointillés, une structure alternative.

Source : Wikimedia http://commons.wikimedia.org/wiki/File:Pseudoknot.svg, modifié par auteur (J. Nicolas)

L'apprentissage d'un automate d'états fini par fusion.



Source : Auteur (J. Nicolas)

Le treillis des partitions d'états correspondant à l'apprentissage d'un langage régulier à partir des phrases {ac, ag}. On a effectué un zoom sur certains éléments du treillis. Pour les autres, on a simplement indiqué les états fusionnés. La double ellipse est le plus grand langage compatible avec l'ensemble de phrases impossibles {a, aag, agg, aca, acg}. Il est constitué des mots formant une suite éventuellement vide de ag terminée par ac (par exemple agagac fait partie du langage).

Pour aller plus loin…

The language of genes, David Searls, nov. 2002. Nature 420 211-217 Version accessible sur ftp://ftp.cis.upenn.edu/pub/cse140/public_html/2002/Searls.pdf

La grammaire de la vie. Antoine Danchin, 2009. http://www.normalesup.org/~adanchin/causeries/grammaire.html

Langages formels, calculabilité et complexité. Olivier Carton, Paris, Vuibert, coll. « Capes-agrég », oct. 2008. Version accessible sur http://www.normalesup.org/~bisson/tea/lfcc.pdf