The language of genes

David B. Searls

Bioinformatics Division, Genetics Research, GlaxoSmithKline Pharmaceuticals, 709 Swedeland Road, PO Box 1539, King of Prussia, Pennsylvania 19406, USA (e-mail: david_b_searls@gsk.com)

Linguistic metaphors have been woven into the fabric of molecular biology since its inception. The determination of the human genome sequence has brought these metaphors to the forefront of the popular imagination, with the natural extension of the notion of DNA as language to that of the genome as the 'book of life'. But do these analogies go deeper and, if so, can the methods developed for analysing languages be applied to molecular biology? In fact, many techniques used in bioinformatics, even if developed independently, may be seen to be grounded in linguistics. Further interweaving of these fields will be instrumental in extending our understanding of the language of life.

The science of linguistics has fully as many facets and fields as biology, and like biology, what may be called its 'modern era' can be traced to the 1950s[1]. The decade that unveiled the structure of DNA also witnessed a revolution in linguistics led by Noam Chomsky, whose work radically diversified the field beyond its then-current focus on simply cataloguing the actual utterances of a language, to exploring the mechanisms by which they are produced. Seeking to identify the universals at the core of all languages, he posited a new, generative form of grammar, or set of syntactic rules, that would help to account for the immense creativity in the production of language that emerges so rapidly as individuals develop[2].

In pursuit of his 'universal grammar', Chomsky created waves that washed up on many scientific shores. Besides his profound influence on theoretical linguistics, his mathematical approach to the description of languages prompted a burst of development in formal language theory. This produced methods with widespread utility in computer science, from the specification and interpretation of computer languages to the fields of syntactic pattern recognition, natural language processing and speech understanding[3]. The Chomsky hierarchy of language classes has proven especially durable as a means of stratifying formal languages according to their expressive power and resulting computational and mathematical complexity (Box 1). Chomsky's influence has also extended to cognitive science, analytic philosophy and even literary criticism. The common experience in a number of fields is that it is not only analytic techniques derived from linguistics, but also what might be called a linguistic sensibility, that can illuminate and inform other similarly complex domains.

Mathematical linguistics and macromolecules

In the 1980s, several workers began to follow various threads of Chomsky's legacy in applying linguistic methods to molecular biology. Early results included the fundamental observation that formal representations could be applied to biological sequences[4] — the extension of linguistic formalisms in new, biologically inspired directions[5] — and the demonstration of the utility of grammars in capturing not only informational but also structural aspects of macromolecules[6].

Nucleic acid linguistics

From this work there followed a series of mathematical results concerning the linguistics of nucleic acid structure[7–9]. These results derive from the fact that a folded RNA secondary structure entails pairing between nucleotide bases that are at a distance from each other in the primary sequence, establishing relationships that in linguistics are called dependencies. The most basic secondary-structure element is the stem-loop, in which the stem creates a succession of nested dependencies that can be captured in idealized form by the following context-free base-pairing grammar[7] (Box 1):

$$S \rightarrow gSc \qquad S \rightarrow cSg \qquad S \rightarrow aSu \qquad S \rightarrow uSa \qquad S \rightarrow \varepsilon$$

(The ε in the last rule indicates that an S is simply erased.) This grammar affords any and every derivation of 'hairpin' sequences of a form such as the following:

$$S \Rightarrow gSc \Rightarrow gaSuc \Rightarrow gauSauc \Rightarrow \ldots$$
$$\ldots \Rightarrow gaucgaSucgauc \Rightarrow gaucgaucgauc$$

Derivations from this grammar grow outward from the central S, creating the nested dependencies of the stem (Fig. 1a), analogous to such phenomena as nested relative clauses in natural language (for example, "The gene that the scientist whom our grant supported discovered encoded a kinase"). In a realistic stem-loop, the derivation would terminate in an unpaired loop of at least several bases and might also contain, for example, non-Watson–Crick base pairs and 'bulges'. But such features are easily added to the grammar without affecting the fundamental result that any language consisting of RNA sequences that fold into these basic structures requires context-free expression[10].

In addition to stem-loop structures, arbitrarily branched folded structures may be captured by simply adding to the grammar above a rule $S \rightarrow SS$, whose application creates bifurcations in the derivation tree[7] (Fig. 1b). The base-pairing dependencies remain non-crossing, although more complicated. The resulting grammar is formally ambiguous, meaning that there are guaranteed to be sequences in the language for which more than one derivation tree is possible[10]. Thus, the string *gaucgaucgauc* can be derived as a single hairpin or as a branched structure (Fig. 1a, b). This linguistic property of ambiguity, reflected in natural languages in sentences that can be syntactically parsed in more than one way (for example, "She saw the man with the telescope"), directly models the biological phenomenon of alternative secondary structure[7]. Although these models are only abstractions of a thermodynamically determined process, ambiguity allows them to embody the ensemble of potential secondary structures, and more specific grammars can specify particular forms, such as transfer RNA cloverleafs[9].

Box 1

The Chomsky hierarchy and formal language theory

Formal language theory defines languages to be nothing more than sets of strings of symbols drawn from some alphabet. A grammar is a rule-based approach to specifying a language, consisting of a set of rewriting rules that take forms such as *A*→*xB*. Here, upper-case letters denote temporary or nonterminal symbols, which do not occur in the alphabet, whereas lower-case letters are terminal symbols that do. The example rule specifies that any occurrence of the nonterminal *A* may be replaced by an *x* followed by a *B*.

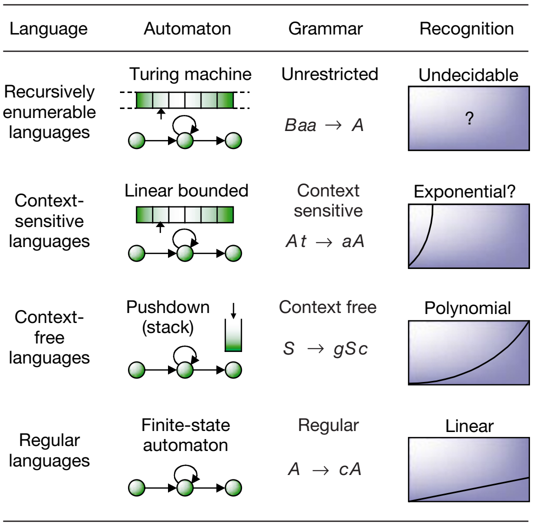
Beginning with a starting nonterminal *S*, a derivation from a grammar consists of a series of rewriting steps that ends when the last nonterminal is eliminated. Consider the simple grammar with an alphabet *x* and *y*, and containing the rules *S*→*xS* and *S*→*y*. This grammar generates all strings beginning with any number of *x*'s and ending in a single *y*. It produces derivations such as *S*⇒*xS*⇒*xxS*⇒*xxxS*⇒*xxxy*, where each double arrow signifies the application of a single-arrow rule. In this case there are three applications of the first rule followed by a single application of the second to produce a terminal string, one of the infinite number of such strings in this language.

Any grammar whose rules rewrite a nonterminal as a terminal followed by at most one nonterminal is called regular, and is said to generate a regular language. An equivalent means of generating such languages is a finite-state automaton (FSA), a notional machine used to reason about computation, built out of states (circles; see figure opposite) which are interconnected by transitions (arrows) that emit symbols from the alphabet as they are traversed.

Grammars that allow any arrangement of terminals and nonterminals on the right-hand sides of rules have greater expressive power. They are called context-free grammars, and can generate not only all regular languages, but also non-regular languages such as strings of *x*'s followed by the same number of *y*'s (for example, *xxxxyyyy*). Such languages cannot be specified by a regular grammar or FSA because these devices have no mechanism for 'remembering' how many *x*'s were generated when the time comes to derive the *y*'s. This shortcoming is remedied by means of context-free rules such as *S*→*xSy*, which always generate an *x* and a *y* at the same time. Alternatively, an automaton augmented with a push-down store, a memory device that pushes or pops symbols to or from a stack during transitions, also provides such a counting capability. In either case, context-free languages allow strings that embody dependencies between terminals, such as the relationship matching *x*'s and *y*'s in the example, provided that those dependencies can be drawn as nested, either strictly within or independent of each other, but never crossing.

Even context-free grammars are inadequate for some languages, for instance strings of consecutive *x*'s, *y*'s and *z*'s in equal number (for example, *xxxyyyzzz*). This entails dependencies that necessarily cross one another, called cross-serial dependencies, and to capture these with a grammar requires rules that have additional symbols on their left-hand side (though never more than on their right-hand side). Such context-sensitive rules correspond to automata with a more sophisticated memory device, a tape whose length is bounded in a certain way, upon which the machine can read and write symbols. Context-sensitive languages include all context-free languages and many more, yet theoretically there exist languages outside even this set, called recursively enumerable languages, generated by grammars of completely unrestricted form or by machines with unbounded tapes best known as Turing machines.

In the figure above, the language classes in the left column contain exactly those languages that can be generated by the automata and grammar types indicated in the next two columns. Each level contains all of those below it. The right-hand column illustrates the computational complexity, in the general case, of recognizing whether a string belongs in a given language, showing how the time required grows as a function of the length of the input string. At the highest level of the hierarchy, one is not even guaranteed to be able to arrive at an answer by computational means. This is just one indication of the trade-off between the increase in expressive power afforded by ascending the Chomsky hierarchy, and the mathematical and algorithmic limitations that invariably result[24].



Language	Automaton	Grammar	Recognition
Recursively enumerable languages	Turing machine	Unrestricted *Baa* → *A*	Undecidable ?
Context-sensitive languages	Linear bounded	Context sensitive *At* → *aA*	Exponential?
Context-free languages	Pushdown (stack)	Context free *S* → *gSc*	Polynomial
Regular languages	Finite-state automaton	Regular *A* → *cA*	Linear

Finding that the language of RNA is at least context-free has mathematical and computational consequences, for example, for the nature and inherent performance bounds of any algorithm dealing with secondary structure (Box 1). For instance, the fast, regular-expression search tools used commonly in bioinformatics (such as those in the popular Perl scripting language) are ruled out, as in their standard form they specify only regular languages. These consequences show the importance of characterizing linguistic domains in the common terminology and methodology of formal language theory, so as to connect them immediately to the wealth of tools and understanding already available. For this reason, recent bioinformatics textbooks have devoted whole chapters to the relationship of biological sequences to the Chomsky hierarchy[11,12].

In light of these practical consequences of linguistic complexity, a significant finding is that there exist phenomena in RNA that in fact raise the language even beyond context-free. The most obvious of these are so-called non-orthodox secondary structures such as pseudoknots, which are pairs of stem-loop elements in which part of one stem resides within the loop of the other (Fig. 1c). This configuration induces cross-serial dependencies in the resulting base pairings, requiring context-sensitive expression (Box 1). Predictably, given this further promotion in the Chomsky hierarchy, the need to encompass pseudoknots within secondary-structure recognition and prediction programs has significantly complicated algorithm design[13]. Another non-context-free phenomenon that occurs in RNA is a consequence of alternative secondary structure, such as that seen in bacterial attenuators, which are regulatory elements that depend on switching between conformations in nascent mRNA molecules. For any grammar required to simultaneously represent both conformations, these mutually exclusive options create overlapping (and thus cross-serial) dependencies in the alternate base-pairing schemes[7] (Fig. 1d).

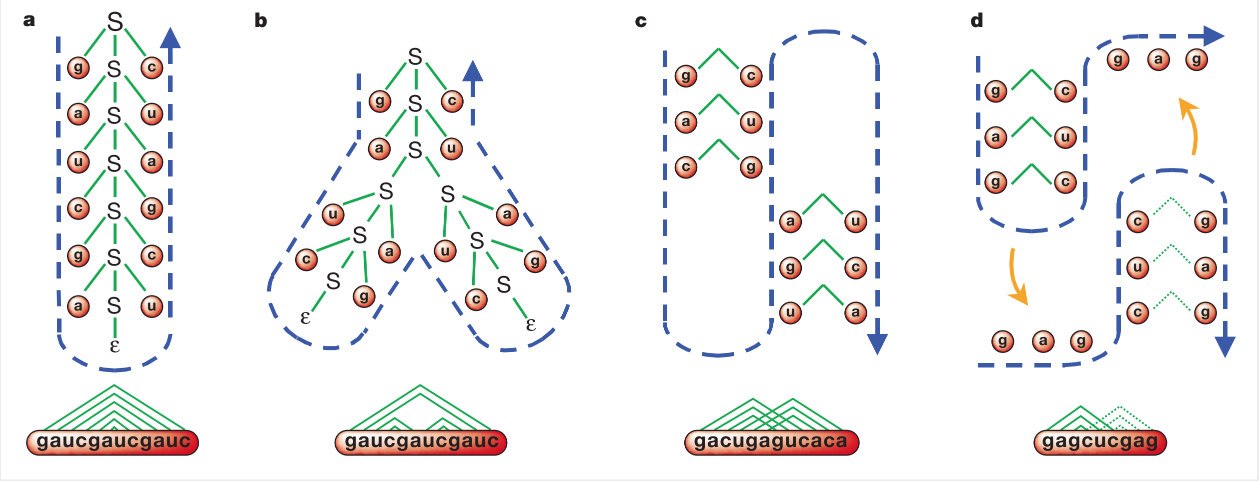


Figure 1 Grammar-style derivations of idealized versions of RNA structures. **a**, A stem; **b**, a branched structure; **c**, a pseudoknot; and **d**, alternative secondary structures of an attenuator. The trees for **a** and **b** are graphical depictions of derivations from grammars given in the text. By convention, a starting nonterminal *S* is at the root of the tree and gives rise to branches for each symbol to which it rewrites in the course of the derivation. The string derived can be read by tracing the frontier or leaf nodes of the tree, left to right (dashed blue lines). For **c** and **d**, derivation trees are not explicitly indicated because of the complexity of the context-sensitive grammars required[7]. The same strings are also shown in linear fashion, with dependencies indicated between terminals derived at the same steps.

Using formalisms called tree-adjoining grammars and their variants[14], which are considered to be mildly context-sensitive and relatively tractable, it is possible to encompass a wide range of RNA secondary structures[15]. Additionally, new types of grammars have been invented to deal with such biological examples[16,17]. Natural languages seem to be beyond context-free as well, based on linguistic phenomena entailing cross-serial dependencies[18], although in both domains such phenomena seem to be less common than nested dependencies. Thus, by one measure at least, nucleic acids may be said to be at about the same level of linguistic complexity as natural human languages.

Protein linguistics

There has been less activity in modelling proteins with linguistic methods, perhaps because they are viewed as having a richer basic repertoire of interactions and conformations than nucleic acids, and perhaps also more of a sense of emergent properties. Yet grammars can be extraordinarily detailed and nuanced (while remaining manageable because of their inherently modular and hierarchical design), and moreover need not capture every aspect of a structure to be useful. In fact, the comprehensiveness and proper role of grammars remains as much an issue for natural language as it might prove to be for proteins, as does the question of whether exemplars of either language are susceptible of a compositional semantics (that is, one for which the meaning or function of the whole can be built up in rule-based fashion from that associated with its parts)[3]. In any case there is a decidedly linguistic flavour to certain abstracted depictions of protein structure, such as domain schematics (for example, the SMART system, which portrays the highly variable arrangements of 'mobile' domains[19]) or topology 'cartoons' (for example, the TOPS system, which annotates dependencies between secondary structural elements, including positional and chiral relationships[20]).

Specific aspects of protein structure have been modelled explicitly with grammars. Secondary structural elements, and in particular the hydrogen bonding between strands in a β-sheet, may be arrayed in antiparallel fashion, creating nested dependencies by analogy with stem-loop structures in RNA, or in parallel fashion, which creates cross-serial dependencies. Such arrangements have been represented using stochastic tree grammars[21], which are related to tree-adjoining grammars and which have also been shown to generate a range of configurations of β-sheets that corresponds well to that seen in nature (A. Joshi, personal communication). Another grammar-based approach, using tools from graph theory, was shown recently to be capable of generating a preponderance of the class of all-β-folds from just four basic rules[22].

Mathematicians are concerned with closure properties of languages, that is, whether they remain at the same level of the Chomsky hierarchy when various operations are performed on their contents[9]. Simple concatenation of strings is a so-called regular operation, whereas insertion of one string in another is a context-free operation, insofar as it never causes dependencies to cross, but only further nests them. Neither operation raises a context-free language beyond context-free, nor (it can be shown) do a series of biological operations such as replication and recombination[10]. However, translocation of segments of a string may create cross-serial dependencies where none existed before, and thus the block movements typical of genomic rearrangements may constitute an upward force in the Chomsky hierarchy that is inherent in evolution[10].

Nevertheless, within proteins we see evidence that at the level of domains (if not supersecondary structure) there is again a relative scarcity of non-context-free forms (Fig. 2). This is perhaps attributable not only to the greater complexity of the genomic changes required, but also to the energetic barriers that might be anticipated in folding knot-like cross-serial dependencies, by analogy with difficulties they pose in linguistic analysis. In light of this, it is interesting that the special case of circular permutations (that is, head-to-tail rearrangements), to which protein domains seem more prone[23], do in fact preserve context-free status from a mathematical perspective[24].

Computational linguistics and genes

The results summarized above all relate to structural aspects of macromolecules, that is, factors inherent in their biophysical behaviour and independent of any information they contain. Yet genes do convey information, and furthermore this information is organized in a hierarchical structure whose features are ordered, constrained and related in a manner analogous to the syntactic structure of sentences in a natural language. It is thus not surprising that a number of themes, both explicit and implicit, have found their way from computational linguistics to computational biology.

One implicit theme is a convergence between organizational schemes in the two fields. Language processing is often conceived as proceeding from (1) the lexical level, at which individual words from

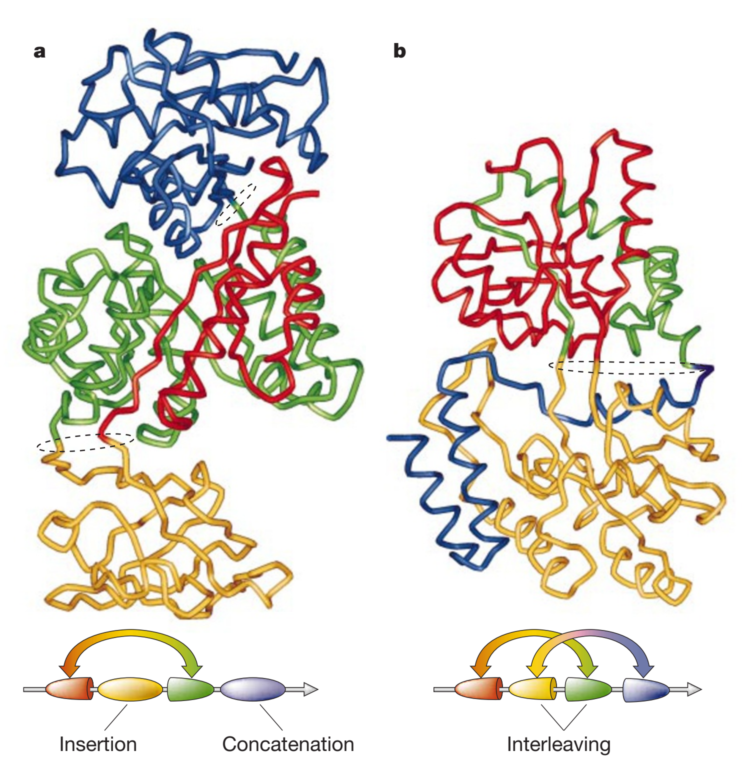


Figure 2 Protein domain arrangements and the Chomsky hierarchy. Shown are backbone structures for **a**, cat muscle pyruvate kinase (1pkm in Protein Data Bank; minus a short amino-terminal domain) and **b**, *Escherichia coli* D-maltodextrin binding protein (1omp in Protein Data Bank). At the bottom are schemas of the domain relationships, with double arrows connecting segments participating in the same domain. The upper, carboxy-terminal (blue) domain of 1pkm attaches by way of a simple concatenation, which is a regular operation commonly seen in proteins. The central red-and-green α/β-barrel, however, is interrupted in the middle by an insertion of the lower (orange) domain, a context-free operation insofar as it thus creates a strictly nested dependency between the divided domain segments (as would any number of domain insertions at any point). Insertions are less common than concatenations, but still fairly frequent. The two main domains of 1omp, on the other hand, seem to be interleaved, thus creating cross-serial dependencies that are necessarily context-sensitive. Whether the C-terminal (blue) segment is involved fully in the lower domain's core, however, is open to question; in any case, true interleaved structural domains seem to be very rare. The dashed ellipses in the backbone diagrams illustrate that the number of crossovers between domains (1, 2 and 3, respectively) is indicative of the level in the Chomsky hierarchy of the resulting domain arrangement.

a linear input stream (of, for example, phonemes or characters) are recognized and characterized; to (2) the syntactic level, at which words are grouped and related hierarchically according to grammar rules to form a structural description; to (3) the semantic level, at which some representation of meaning is assigned to the resulting structure, derived from that of its individual lexical elements; and finally to (4) the pragmatic level, at which language is viewed in a larger context encompassing the roles and interrelationships of sentences (and certain references within them such as pronouns) in an overall discourse or dialogue[3]. This progression maps neatly and meaningfully onto one used widely in biology, of sequence to structure to function to role[25].

In particular, the distinction between syntax and semantics (famously exemplified by Chomsky with his grammatical yet meaningless "Colourless green ideas sleep furiously"[2]) is pertinent to biology. Consider two types of sequence: a string of words, and a segment of a genome. A parsing step may be seen as determining whether the words form a grammatical sentence, or, notionally, whether the genomic sequence will support the production of a polypeptide according to rules implicit in the transcriptional and translational machinery of the cell; in both cases the processes are mechanical, in fact largely processive. Then, an interpretative step determines whether the resulting sentence is meaningful, according to laws of logic and experience, or whether the polypeptide will fold into a compact core and orient its side chains so as to do useful work, a process governed by laws of thermodynamics and biochemistry. Mutated genes that are expressed but do not allow for a functional fold may be said to pass the first test but not the second.

The natural history of gene-finding algorithms offers another illustration. In the 1980s, detecting genes (in what genomic sequence was then extant) was strictly a lexical affair. Algorithms simply scanned an input sequence and within a moving window assessed its 'coding potential' on the basis of statistical measures such as oligonucleotide frequencies and periodicities. It was also possible to detect signals such as putative splice sites, again as individual lexical elements. Then, in the early 1990s, programs began to appear that assembled lexical elements hierarchically and imposed constraints of a distinctly syntactic cast. (Thus, just as sentence constituents must agree as to number, gender, tense, and so on, so had putative exons to maintain a reading frame across whole genes.) Indeed, one program that performed creditably at the time was based explicitly on a gene grammar and a general-purpose parser (a program that determines if an input is a valid instance of any given grammar and, if so, produces a tree-structured description of the parse)[26].

One advantage of linguistic gene recognition was the natural accommodation of ambiguity in the form of multiple transcripts attributable, for example, to alternative splicing. Another advantage was versatility: the same parser, but with different grammars substituted, was effective in recognizing such features as tRNA genes and group I introns, including secondary structure extending to pseudoknots[27]. Yet another area in which grammars have proven apt is in the specification of gene regulatory elements, with their highly variable distribution of disparate features. This use, in fact, was one of the first suggested biological applications of Chomsky-style grammars[28] and remains an active area of research[29,30].

Although having the advantage of flexibility, general-purpose parsers cannot compete in efficiency with programming that is customized to a particular domain, especially one that does not greatly benefit from the capacity of grammars to specify variations on a theme with ease. (English grammar would be superfluous if every sentence were patterned on the same basic declarative template.) Consequently, latter-day gene-finding algorithms, which have the 'standard model' gene structure hard-wired, do not make use of grammars *per se*. However, what has instead become a dominant technique in the analysis of biological sequences, the hidden Markov model (HMM), also traces its pedigree to linguistic roots and inherits a different set of advantages.

An HMM is a variety of automaton annotated with probability values that govern its behaviour[3]. They were first widely deployed in the field of speech recognition and more recently have found their way into a number of applications for the analysis of biological sequences, beginning with protein family profiles[11]. HMM

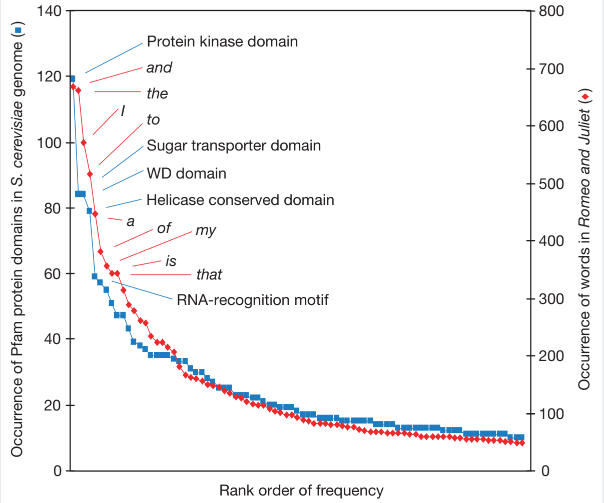


Figure 3 Distributions of the number of occurrences of Pfam protein domains (blue squares) in the genome of the yeast *Saccharomyces cerevisiae*, and of words (red diamonds) in Shakespeare's *Romeo and Juliet*, in both cases sorted in rank order from left to right. The most frequently occurring domains and words are labelled. In both cases (and in many other genomes and texts) the curves are good fits to a power-law distribution known as Zipf's law, which relates the frequency to the inverse of the rank.

architectures embody what amounts to a syntax and use an associated set of algorithms to refine and employ the model. HMMs with sophisticated domain models form the basis for several leading gene finders, including GenScan[31] and Genie[32], and the gene-finding application has driven further refinement of the method as well. The recent marked trend in computational biology towards probabilistic methods such as HMMs has mirrored a similar turn in natural language processing, which has been invigorated by a shift towards finite-state and stochastic approaches[3]. The use of HMMs in the two fields has been compared directly in a recent review[33].

The automata associated with HMMs are at the lowest rung of the Chomsky hierarchy and are thus inadequate for such non-regular features as the secondary structure in tRNA. This shortcoming has been addressed by adding probabilities to context-free grammars to create stochastic context-free grammars and then adapting the HMM algorithms to work with the resulting data structures[34]. Such systems have proven useful not only in tRNA detection[35], but also in a variety of related biological applications[36–39], and have even been extended to non-context-free structures[40].

Historical linguistics and evolution

Long before Chomsky's revolution, historical linguistics was the dominant discipline in the field[41], driven largely by an increasingly systematic attempt to account for the descent of modern languages from a hypothesized proto-Indo-European language first proposed in 1786[1]. Of this work Darwin himself noted that "the formation of different languages and of distinct species, and the proofs that both have been developed through a gradual process, are curiously parallel"[42]. These parallels have since inspired many authors. Dawkins' concept of 'memes' as replicating cultural fragments undergoing darwinian selection encompasses language change[43], as does a recent synthesis of formal language theory, learning theory and evolutionary dynamics[44]. Strong analogies between the evolution of languages and of species have even formed the basis for serious scientific arguments against creationism[45]. Cavalli-Sforza has comprehensively explored how population genetics can aid understanding of language evolution from a demographic perspective[46], and biological phylogenetic-reconstruction techniques have also been applied to languages[47].

Among the methods linguists themselves have used to draw 'family trees' of languages has been the statistical comparison of vocabularies, or lexicostatistics[41]. This approach posits that, across many languages, there is a basic, core set of cognates (essentially, word 'orthologues') relating to universal human experience and relatively resistant to change. In the 1950s, Swadesh established 200 such concepts (for example, *I*, *this*, *not*, *person*, *fish*, *blood*, *egg*, *knee*, *cloud*, *mountain* and *good*) and, based on similarity of corresponding words in different languages, derived quantitative measures of overall language relatedness[48]. He further proposed that language divergences could be dated in this manner by assuming a constant rate of lexical change, a technique called glottochronology. Although controversial, this is clearly echoed in the notion of the evolutionary 'molecular clock'. Indeed, the need to account for varying rates of change in different words and proteins has been recognized independently in each field[49].

The compilation of core vocabularies from multiple languages resembles efforts to assemble 'minimal gene sets' presumed sufficient to support life (by one estimate, numbering about 300) by taking intersections of multiple genomes, and similar cautions have been noted in their use and interpretation[50]. For instance, from the fact that French has no word for 'shallow' one could not conclude that the language is impoverished, any more than the apparent absence of a given enzyme necessarily rules out a certain metabolic capacity. Comparisons of gene contents across phylogeny have been used in ways that might have been drawn directly from the lexicostatistical literature. Examples include the collection of clusters of orthologous groups[51] and the use of degree of overlap of gene complements (as opposed to individual sequence similarities) as a basis for phylogeny construction[52–54] as well as a predictor of protein function[55]. Both fields contend with complications introduced by synonyms and false cognates ('*faux amis*') on the one hand, and on the other, non-orthologous gene displacement and functional shifts, while recent theory concerning reticulate evolution harkens back to well-studied phenomena of language mixture such as creolization[56].

Words themselves arise and evolve by mechanisms that have been compared to biological drivers of diversity, such as mutation and recombination (called blending by linguists)[57]. One mechanism they clearly have in common is compounding. The atomic units of linguistic meaning are morphemes, typically stems and affixes that combine to form words, whereas lexical units are lexemes, which may be single words or compounds and certain unitary phrases[3]. In like manner, proteins are considered to comprise one or more functional domains, and a recent study hypothesizes ancient 'antecedent domain segments', relating these explicitly to linguistic variation[58].

There is more than a surface similarity to such conventions, insofar as these are all elements that are surmised to combine and re-assort in the course of evolution, affording combinatorial diversity, and some of the same techniques have been applied in their analysis. For instance, a quantitative approach to the association of words is collocation analysis. Here, the frequency of co-occurrence of words in text is not only a useful heuristic in stochastic parsing[3], but also provides clues in lexical semantic studies, for which compounds have been classified into such categories as noun+noun constituents, idioms, and so forth[59]. This technique has been 'reinvented' in the counting of gene fusions across many genomes as a predictor, for example, for protein–protein interactions or participation of proteins in common pathways[60]. In both cases, practical implementations call for such steps as filtering of 'promiscuous' elements that are less predictive of common function or meaning[61].

Literary linguistics and the genome

What might be called 'literary linguistics' includes pursuits ranging from stylistics to textual analysis to literary criticism. Although seemingly at opposite poles from the 'hard science' of molecular biology, these activities are at some level not so different from the increasingly hermeneutic role of the bioinformatician, insofar as both are concerned with comparing texts, detecting subtle patterns and relationships, and elucidating theme and variation[25]. Nor is textual criticism devoid of quantitative methods; concern with issues such as authorship attribution and authenticity has engendered an active discipline of statistical literary studies aided by computing[62,63].

The most pervasive theme in all such work is the study of word frequencies in texts, the mathematical analysis of which originates with the linguist G. K. Zipf, who first observed a power-law distribution relating a word's frequency of occurrence to the inverse of its position in the rank ordering of those frequencies[64] (Fig. 3). Mandelbrot elaborated on this insight, proposing a relationship between what has come to be known as Zipf's law and a presumed fractal nature of languages[65]. Apparent instances of power-law behaviour have now been observed in many facets of molecular biology, including oligonucleotide frequencies[66], sizes of gene families[67] (including pseudogenes[68]), distributions of protein[69] and RNA[70] folds, and even levels of gene expression[71]. As in the linguistic case, several explanations for these power-law behaviours have been proposed, including their mathematical relationship to scale-free networks[72] such as might be expected in metabolic pathways[73] and protein interaction maps[74], and models for how they might arise in the evolution of protein families[69], all of which evince comparison to properties of words.

Textual criticism shares both goals and methods with bioinformatics. Species-specific distributions of oligonucleotides are among the signals (called style markers by linguists[62]) that have been used in 'authorship attribution' of genome segments thought to arise by horizontal transmission between species (for example, pathogenicity islands in bacteria[75]), and in checking the 'authenticity' of cloned sequences possibly contaminated by foreign material[76]. Word frequencies and many other style markers have been analysed in literature using such tools as clustering[77], principal components analysis[78], neural networks[79], support vector machines[80] and genetic algorithms[81], all of which are now being applied as well to 'transcript frequencies' inferred from microarray experiments. A recent review of these methods applied to gene expression comes full circle by using a clustering algorithm to group and classify articles on the topic based on word frequencies[82], a foray into what has been termed bibliomics[33,83].

The complexity of human and biological-sequence languages at a lexical level has been compared explicitly by Trifonov and co-workers[84]. Using metrics designed to detect the extent of 'overlapping codes', they suggest that sequence languages are more layered, with multiple signals reflecting, for example, different cellular processes, and thus more 'complex' insofar as the codes may constrain or interfere with one another[85]. (Extreme examples are viral genomes with overlapping, frameshifted coding regions.) It should be noted, however, that human language is not 'single code' as suggested by Trifonov, but involves layering at multiple levels. An obvious illustration is poetry, where lexical and syntactic accommodations are often made for such overlaid constraints as rhyme scheme, metre and verse form, and even higher orders of metaphor, mood and theme — witness the virus-like economy of a *haiku*. Such superposition in languages is even treated formally, insofar as context-free languages are not closed under intersection and thus may be driven higher in the Chomsky hierarchy by layering[10]; a specific instance is the view of a pseudoknot as the intersection of two stem-loop structures[40].

A branch of textual criticism called stemmatics is concerned with the accuracy of texts, possibly ancient, that exist in multiple forms for reasons ranging from printers' errors to authorial revisions to fragmentary sources. For manuscripts copied many times by scribes, there has even been mathematical modelling of copying errors for purposes of estimating pairwise distances along a path from a common ancestor[86]; biologically motivated algorithms have been enlisted in this cause to elucidate the provenance of Chaucer's *Canterbury Tales*[87]. However, the very foundation of these algorithms in biological cladistics recapitulates older, similar methods from stemmatics and linguistics, as was already recognized a quarter-century ago[88].

One post-modern (and thus antiauthoritarian) school of textual criticism promotes the idea of a genetic text, a dynamic concept that encompasses all versions and even sources of a text through time[89], largely abandoning the concept of a 'main' version and thereby requiring new organizational paradigms and computational aids[90]. The genetic text that is the genome surely presents similar challenges, and the many commonalities (as well as the instructive differences) between natural and biological languages may thus form the basis for sharing tools, techniques and ways of thinking about complex systems, on many different levels. □

doi:10.1038/nature01255

1. Aitchison, J. *Linguistics* (NTC/Contemporary Publishing, Chicago, 1999).
2. Chomsky, N. *Syntactic Structures* (Mouton, The Hague, 1957).
3. Jurafsky, D. & Martin, J. H. *Speech and Language Processing* (Prentice Hall, Upper Saddle River, NJ, 2000).
4. Brendel, V. & Busse, H. G. Genome structure described by formal languages. *Nucleic Acids Res.* **12,** 2561–2568 (1984).
5. Head, T. Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bull. Math. Biol.* **49,** 737–759 (1987).
6. Searls, D. B. in *Proc. 7th Natl Conf. Artif. Intell.* 386–391 (AAAI Press, Menlo Park, CA, 1988).
7. Searls, D. B. The linguistics of DNA. *Am. Sci.* **80,** 579–591 (1992).
8. Searls, D. B. in *Logic Programming: Proc. North Am. Conf.* (eds Lusk, E. & Overbeek, R.) 189–208 (MIT Press, Cambridge, MA, 1989).
9. Searls, D. B. in *Artificial Intelligence and Molecular Biology* Ch. 2 (ed. Hunter, L.) 47–120 (AAAI Press, Menlo Park, CA, 1993).
10. Searls, D. B. in *Mathematical Support for Molecular Biology* (eds Farach-Colton, M., Roberts, F. S., Vingron, M. & Waterman, M.) 117–140 (American Mathematical Society, Providence, RI, 1999).
11. Durbin, R., Krogh, A., Mitchison, G. & Eddy, S. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (Cambridge Univ. Press, Cambridge, 1998).
12. Baldi, P. & Brunak, S. *Bioinformatics: The Machine Learning Approach* (MIT Press, Cambridge, MA, 2001).
13. Lyngso, R. B. & Pedersen, C. N. RNA pseudoknot prediction in energy-based models. *J. Comput. Biol.* **7,** 409–427 (2000).
14. Joshi, A. in *Natural Language Processing: Psycholinguistic, Computational and Theoretical Perspectives* (eds Dowty, D., Karttunen, L. & Zwicky, A.) 206–250 (Chicago Univ. Press, New York, 1985).
15. Uemura, Y., Hasegawa, A., Kobayashi, S. & Yokomori, T. Tree-adjoining grammars for RNA structure prediction. *Theor. Comput. Sci.* **10,** 277–303 (1999).
16. Searls, D. B. String Variable Grammar: a logic grammar formalism for DNA sequences. *J. Logic Program.* **24,** 73–102 (1995).
17. Rivas, E. & Eddy, S. R. The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics* **16,** 334–340 (2000).
18. Shieber, S. Evidence against the context-freeness of natural language. *Linguist. Phil.* **8,** 333–343 (1985).
19. Schultz, J., Milpetz, F., Bork, P. & Ponting, C. P. SMART, a simple modular architecture research tool: identification of signalling domains. *Proc. Natl Acad. Sci. USA* **95,** 5857–5864 (1998).
20. Westhead, D. R., Slidel, T. W., Flores, T. P. & Thornton, J. M. Protein structural topology: automated analysis and diagrammatic representation. *Protein Sci.* **8,** 897–904 (1999).
21. Abe, N. & Mamitsuka, H. Predicting protein secondary structure using stochastic tree grammars. *Machine Learn.* **29,** 275–301 (1997).
22. Przytycka, T., Srinivasan, R., & Rose, G. D. Recursive domains in proteins. *Protein Sci.* **11,** 409–417 (2002).
23. Jung, J. & Lee, B. Circularly permuted proteins in the protein structure database. *Protein Sci.* **10,** 1881–1886 (2001).
24. Hopcroft, J. E. & Ullman, J. D. *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, MA, 1979).
25. Searls, D. B. Reading the book of life. *Bioinformatics* **17,** 579–580 (2001).
26. Dong, S. & Searls, D. B. Gene structure prediction by linguistic methods. *Genomics* **23,** 540–551 (1994).
27. Searls, D. B. Linguistic approaches to biological sequences. *Comput. Appl. Biosci.* **13,** 333–344 (1997).
28. Collado-Vides, J. A transformational-grammar approach to the study of the regulation of gene expression. *J. Theor. Biol.* **136,** 403–425 (1989).
29. Rosenblueth, D. A. *et al.* Syntactic recognition of regulatory regions in *Escherichia coli*. *Comput. Appl. Biosci.* **12,** 15–22 (1996).
30. Leung, S., Mellish, C. & Robertson, D. Basic Gene Grammars and DNA-ChartParser for language processing of *Escherichia coli* promoter DNA sequences. *Bioinformatics* **17,** 226–236 (2001).
31. Burge, C. & Karlin, S. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268,** 78–94 (1997).
32. Reese, M. G., Kulp, D., Tammana, H. & Haussler, D. Genie—gene finding in *Drosophila melanogaster*. *Genome Res.* **10,** 529–538 (2000).
33. Yandell, M. D. & Majoros, W. H. Genomics and natural language processing. *Nature Rev. Genet.* **3,** 601–610 (2002).
34. Sakakibara, Y. *et al.* Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res.* **22,** 5112–5120 (1994).
35. Lowe, T. M. & Eddy, S. R. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.* **25,** 955–964 (1997).
36. Rivas, E. & Eddy, S. R. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* **2,** 8 (2001).
37. Knudsen, B. & Hein, J. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics* **15,** 446–454 (1999).
38. Brown, M. P. Small subunit ribosomal RNA modeling using stochastic context-free grammars. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8,** 57–66 (2000).
39. Holmes, I. & Rubin, G. M. Pairwise RNA structure comparison with stochastic context-free grammars. *Pac. Symp. Biocomput.* 163–174 (2002).
40. Brown M. & Wilson C. RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. *Pac. Symp. Biocomput.* 109–125 (1996).

41. Campbell, L. *Historical Linguistics: An Introduction* (MIT Press, Cambridge, MA, 1999).
42. Darwin, C. *The Descent of Man* (John Murray, London, 1871).
43. Dawkins, R. *The Selfish Gene* (Oxford Univ. Press, Oxford, 1976).
44. Nowak, M. A., Komarova, N. L. & Niyogi, P. Computational and evolutionary aspects of language. *Nature* **417,** 611–617 (2002).
45. Pennock, R. T. *Tower of Babel: The Evidence against the New Creationism* (Bradford/MIT Press, Cambridge, MA, 1999).
46. Cavalli-Sforza, L. L. *Genes, Peoples, and Languages* (North Point Press, New York, 2000).
47. Warnow T. Mathematical approaches to comparative linguistics. *Proc. Natl Acad. Sci. USA* **94,** 6585–6590 (1997).
48. Swadesh, M. Lexicostatistical dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos. *Proc. Am. Phil. Soc.* **96,** 452–463 (1952).
49. Kruskal, J. B., Dyen, I. & Black, P. in *Lexicostatistics in Genetic Linguistics* (ed. Dyen, I.) 30–55 (Mouton, The Hague, 1973).
50. Mushegian, A. The minimal genome concept. *Curr. Opin. Genet. Dev.* **9,** 709–714 (1999).
51. Tatusov, R. L., Galperin, M. Y., Natale, D. A. & Koonin, E. V. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.* **28,** 33–36 (2000).
52. Snel, B., Bork P, & Huynen, M. A. Genome phylogeny based on gene content. *Nature Genet.* **21,** 108–110 (1999).
53. Tekaia, F., Lazcano, A., & Dujon, B. The genomic tree as revealed from whole proteome comparisons. *Genome Res.* **9,** 550–557 (1999).
54. Lin, J. & Gerstein, M. Whole-genome trees based on the occurrence of folds and orthologs: implications for comparing genomes on different levels. *Genome Res.* **10,** 808–818 (2000).
55. Pellegrini, M. *et al.* Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA* **96,** 4285–4288 (1999).
56. McWhorter, J. H. *The Power of Babel: A Natural History of Language* 128–129 (Freeman, New York, 2001).
57. Searls, D. B. From *Jabberwocky* to genome: Lewis Carroll and computational biology. *J. Comp. Biol.* **8,** 339–348 (2001).
58. Lupas, A. N., Ponting, C. P. & Russell, R. B. On the evolution of protein folds: are similar motifs in different protein folds the result of convergence, insertion, or relics of an ancient peptide world? *J. Struct. Biol.* **134,** 191–203 (2001).
59. McKeown, K. R. & Radev, D. R. in *A Handbook of Natural Language Processing* (eds Dale, R., Moisl, H. & Somers, H.) 507–523 (Dekker, New York, 2000).
60. Marcotte, E. M. *et al.* Detecting protein function and protein-protein interactions from genome sequences. *Science* **285,** 751–753 (1999).
61. Smadja, F. Retrieving collocations from text: XTRACT. *Comput. Linguist.* **19,** 143–177 (1993).
62. Rudman, J. The state of authorship attribution studies: some problems and solutions. *Comput. Humanities* **31,** 351–365 (1998).
63. Barnbrook, G. *Language and Computers* (Edinburgh Univ. Press, Edinburgh, 1996).
64. Zipf, G. K. *Human Behavior and the Principle of Least Effort* (Addison-Wesley, Boston, MA, 1949).
65. Mandelbrot, B. *The Fractal Geometry of Nature* (Freeman, San Francisco, 1983).
66. Mantegna, R. N. *et al.* Linguistic features of noncoding DNA sequences. *Phys. Rev. Lett.* **73,** 3169–3172 (1994).
67. Huynen, M. A. & van Nimwegen, E. The frequency distribution of gene family sizes in complete genomes. *Mol. Biol. Evol.* **15,** 583–589 (1998).
68. Harrison, P. M. & Gerstein, M. Studying genomes through the aeons: protein families, pseudogenes and proteome evolution. *J. Mol. Biol.* **318,** 1155–1174 (2002).
69. Qian, J., Luscombe, N. M. & Gerstein, M. Protein family and fold occurrence in genomes: power-law behaviour and evolutionary model. *J. Mol. Biol.* **313,** 673–681 (2001).
70. Schuster, P., Fontana, W., Stadler, P. F. & Hofacker, I. L. From sequences to shapes and back: a case study in RNA secondary structures. *Proc. R. Soc. Lond. B* **255,** 279–284 (1994).
71. Hoyle, D. C., Rattray, M., Jupp, R. & Brass, A. Making sense of microarray data distributions. *Bioinformatics* **18,** 576–584 (2002).
72. Rzhetsky, A. & Gomez, S. M. Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome. *Bioinformatics* **17,** 988–996 (2001).
73. Jeong, H. *et al.* The large-scale organization of metabolic networks. *Nature* **407,** 651–654 (2000).
74. Park, J., Lappe, M. & Teichmann, S. A. Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertoires in the PDB and yeast. *J. Mol. Biol.* **307,** 929–938 (2001).
75. Garcia-Vallve, S., Romeu, A. & Palau, J. Horizontal gene transfer in bacterial and archaeal complete genomes. *Genome Res.* **10,** 1719–1725 (2000).
76. White, O. *et al.* A quality control algorithm for DNA sequencing projects. *Nucleic Acids Res.* **21,** 3829–3838 (1993).
77. Hoover, D. I. Statistical stylistics and authorship attribution: an empirical investigation. *Lit. Linguist. Comput.* **16,** 421–444 (2001).
78. Binongo, J. N. G. & Smith, M. W. A. The application of principal component analysis to stylometry. *Lit. Linguist. Comput.* **14,** 445–466 (1999).
79. Hoorn, J. F., Frank, S. L., Kowalczyk, W. & van der Ham, F. Neural network identification of poets using letter sequences. *Lit. Linguist. Comput.* **14,** 311–338 (1999).
80. Leopold, E. & Kindermann, J. Text categorization with support vector machines. How to represent texts in input space? *Machine Learn.* **46,** 423–444 (2002).
81. Holmes, D. I. & Forsyth, R. S. The Federalist revisited: new directions in authorship attribution. *Lit. Linguist. Comput.* **10,** 111–127 (1995).
82. Altman, R. B. & Raychaudhuri, S. Whole-genome expression analysis: challenges beyond clustering. *Curr. Opin. Struct. Biol.* **11,** 340–347 (2001).
83. Searls D. B. Mining the bibliome. *Pharmacogenomics J.* **1,** 88–89 (2001).
84. Popov, O., Segal, D. M. & Trifonov, E. N. Linguistic complexity of protein sequences as compared to texts of human languages. *Biosystems* **38,** 65–74 (1996).
85. Trifonov, E. N. Interfering contexts of regulatory sequence elements. *Comput. Appl. Biosci.* **12,** 423–429 (1996).
86. Spenser, M. & Howe, C. Estimating distances between manuscripts based on copying errors. *Lit. Linguist. Comput.* **16,** 467–484 (2001).
87. Barbrook, A. C., Howe, C. J., Blake, N. & Robinson, P. The phylogeny of the *Canterbury Tales*. *Nature* **394,** 839 (1998).
88. Platnick, N. I. & Cameron, H. D. Cladistic methods in textual, linguistic, and phylogenetic analysis. *Syst. Zool.* **26,** 380–385 (1977).
89. Tanselle, G. T. *Literature and Artifacts* (Bibliographical Society of the University of Virginia, Charlottesville, VA, 1998).
90. Ferrer, D. Hypertextual representation of literary working papers. *Lit. Linguist. Comput.* **10,** 143–145 (1995).

Acknowledgements

I thank P. Agarwal, A. Lupas, N. Odendahl and K. Rice for helpful comments on the manuscript.