



Springer

Dear Author:

Please find attached the final pdf file of your contribution, which can be viewed using the Acrobat Reader, version 3.0 or higher. We would kindly like to draw your attention to the fact that copyright law is also valid for electronic products. This means especially that:

- You may not alter the pdf file, as changes to the published contribution are prohibited by copyright law.
- You may print the file and distribute it amongst your colleagues in the scientific community for scientific and/or personal use.
- You may make an article published by Springer-Verlag available on your personal home page provided the source of the published article is cited and Springer-Verlag is mentioned as copyright holder. You are requested to create a link to the published article in LINK, Springer's internet service. The link must be accompanied by the following text: The original publication is available on LINK **<http://link.springer.de>**. Please use the appropriate URL and/or DOI for the article in LINK. Articles disseminated via LINK are indexed, abstracted and referenced by many abstracting and information services, bibliographic networks, subscription agencies, library networks and consortia.
- You are not allowed to make the pdf file accessible to the general public, e.g. your institute/your company is not allowed to place this file on its homepage.
- Please address any queries to the production editor of the journal in question, giving your name, the journal title, volume and first page number.

Yours sincerely,

Springer-Verlag Berlin Heidelberg

Junhyong Kim

Descartes' fly: the geometry of genomic annotation

Received: 14 August 2000 / Accepted: 15 August 2000 / Published online: 19 October 2000
© Springer-Verlag 2000

Abstract The completion of the *Drosophila melanogaster* genome marks another significant milestone in the growth of sequence information. But it also contributes to the ever-widening gap between sequence information and biological knowledge. One important approach to reducing this gap is theoretical inference through computational technologies. Many computer programs have been designed to annotate genomic sequence information with biologically relevant information. Here, I suggest that all of these methods have a common structure in which the sequence fragments are “coordinated” by some method of description such as Hidden Markov models. The key to the algorithms lies in constructing the most efficient set of coordinates that allow extrapolation and interpolation from existing knowledge. Efficient extrapolation and interpolation are produced if the sequence fragments acquire a natural geometrical structure in the coordinated description. Finding such a coordinate frame is an inductive problem with no algorithmic solution. The greater part of the problem of genomic annotation lies in biological modeling of the data rather than in algorithmic improvements.

Keywords Gene discovery · Genomic annotation · Hidden Markov models · Motifs

Introduction

This year the (almost) complete sequence of the *Drosophila melanogaster* genome was announced. This marks another significant milestone in our exploding ability to generate sequence information. Yet, it also marks another step in the ever-widening gap between se-

quence information and biological knowledge associated with the sequences (Boguski 1999). The two inroads into increasing biological knowledge are high-throughput systematic experimentation as represented by functional genomics, and theoretical inference as represented by computational biology and bioinformatics. We might broadly use the term data-mining to refer to the various activities surrounding computational approaches to relating biological information to sequence data. Genome projects are somewhat like putting all the library books on a CD-ROM – it alleviates all the physical access problems. However, problems remain because the data from a genome project is similar to having all the books on a CD-ROM without any titles, annotations, cross-references, etc. Data-mining genomes might be analogized to finding a particular sentence in a book in a very large library. “Find a particular sentence” is a purposefully vague statement. This might include searching by patterns like “find a sentence that starts with ‘To be or not to be’”, by syntax such as “find a sentence that starts with a preposition, present subjective verb,...”, or by semantics such as “find a sentence that expresses the angst of a young prince”. All of these search clauses happen in real experimental settings. We may want to identify a gene in silico by its DNA pattern “ACCAGTC...”, by its structure, say seven transmembrane protein, or by phenotype, say olfaction. As in library searches, each of these problems is successively harder. And, some cruel person wrote our CD-ROM in a long-lost language without punctuation or delimiters.

But data-mining is a broad term that means different things in different contexts. For example, it may refer to synthesizing data from heterogeneous databases, automatic text retrieval, or gene discovery, as well as annotating the output of a genomic sequencing project. Problems such as distributed database interoperation or semantic interpretation by a machine are serious and difficult issues. For this review, I will concentrate mainly on the somewhat more straightforward problem of annotating the output of a large-scale sequencing project like the *Drosophila* genome project with biologically useful in-

J. Kim (✉)
Department of Ecology and Evolutionary Biology,
Department of Molecular, Cellular, and Developmental Biology,
Department of Statistics, Yale University, P.O. Box 208106,
New Haven, CT 06520–8106, USA
e-mail: junhyong.kim@yale.edu
Tel.: +1-203-4329917, Fax: +1-203-4323854

formation [e.g. Krogh 2000; Reese et al. 2000 and other related articles in *Genome Research* Vol. 10(4)]. Many different algorithms and methods have been proposed for identifying biologically relevant features in genomic sequences. The purpose of this review is not to examine each of the methods and I refer the reader to other reviews on the subject (e.g. Agarwal and States 1998; Bork 2000; Fickett 1996; Fickett and Wasserman 2000; Singh 2000). Rather, I concentrate on developing a broad theoretical overview of genomic annotation. I will first associate genomic annotation to the more general idea of measurement and classification. I will then review some of the existing approaches to annotation from the point of measurement and classification. Next I will argue that all of the approaches fall into a basic structure of constructing a set of measurements that place sequence fragments in the right “space” of description. Within the right measurement space, sequence fragments fall into geometrically distinct regions that allow us to differentiate among biological categories. An annotation set can be ontologically simple, say exon/intron, or complex, say “house-keeping genes”. In this sense, genomic annotation mirrors the general problem of biological systematics and classification, where the goal is to obtain a description that reveals the “most natural structure” among biological objects. Thus, I connect genomic annotation to geometry and to theories of classification.

Genomic annotation refers to systematic applications of feature identification. Rather than just looking for, say, G protein-coupled receptors in the database, we systematically identify all categorical information of interest. Once annotation is complete, searching for a particular gene or molecule type becomes trivial. (Annotation has been also used in the more narrow sense of predicting the presence of a translated stretch of DNA and its exon-intron structure. Here, I use it in the wider sense of attaching any meaningful labels to fragments of DNA or amino acid.) First, I define the notion of genomic annotation more precisely. Let G be the genomic sequence data, say the entire assembled *D. melanogaster* genome. I will denote S^G as the set of all possible subsets of G . That is, S^G is the collection of all possible sequence fragments. Then the genome annotation problem is that of constructing an annotation map,

$$a_L: S^G \rightarrow L \quad (1)$$

where L is some label set. (A “map” is an assignment between one set of items, say the sequence fragments, to another set of items, say labels.) An example of the label set might be {exon, intron, inter-genic sequence} or {G protein-coupled receptors, non-G protein-coupled receptors}. The label set is the set of possible annotations where we understand the biological meaning of the categories such as “exon” and “intron.” (Of course, the label set need not be strictly categorical but may be more general, such as real values representing genetic map positions or biochemical constants.) Each annotation map is a particular kind of labeling and a complete annotation would involve a collection of such maps. Strictly speak-

ing, saying there is a labeling function α_L is like saying we will use an ostensive definition of objects where one points to objects and says “that is a dog, that is also a dog, that is not a dog...”. As might be imagined, such a procedure is neither operational nor useful.

A popular anecdote is that Descartes invented the Cartesian coordinate system by watching a fly on the ceiling and wondering about the best way to describe its exact position. Notions of positions or geometry existed before Descartes, but his description particularly revealed the natural relationship between algebra and geometry, and the Cartesian coordinate system became an extremely convenient “handle” by which to refer to positions in space. For example, if we wanted to categorize positions as “east” and “west”, we can conveniently say all positions with the X-coordinate greater than 0 are “east” and those with the X-coordinate less than 0 are “west.” Similarly, given the sequence fragments, S^G , there is a need to generate a set of handles to conveniently refer to particular fragments. All genomic annotation procedures implicitly or explicitly create a set of handles on S^G . A natural set of handles on sequences might be a set of measurements related to biological function such as their length, amino acid composition, symmetry, and so on. But more common handles are constructions like the presence/absence of a particular sequence motif.

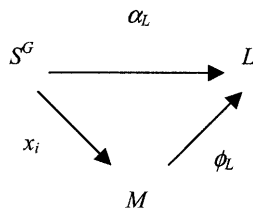
The Cartesian coordinate system for ceiling positions attaches a set of numbers to each position. Similarly, the handles for the sequences attach measurement values to each sequence fragment. In the Cartesian system, a ceiling position is characterized by two numbers, the x and y coordinates. This can be described in a slightly more abstract way by saying that we have two functions, the x -coordinate function and the y -coordinate function, over the ceiling. At any particular position, the value of the coordinate functions gives us two numbers that characterize the position. Similarly, for sequences, a particular handle such as GC content or presence of a sequence motif such as “TATAA” can be seen as a coordinate function on the set of possible sequence fragments and any particular sequence can be characterized by the values of such functions. Therefore, I will use the term coordinate function to refer to a map that assigns to each sequence fragment a number or a value, and use the notation:

$$x_i: S^G \rightarrow R \quad (2)$$

to refer to the i th coordinate function. The word “coordinate” in this sense has natural correspondence to the common-sense use of the word: that is, it is an ordered set of numbers or values that position (characterize) sequence fragments. In Descartes’ ceiling system, two coordinate functions were needed to characterize all positions. When the collection of objects has complex ontology such as in biological sequence fragments, more than one coordinate function is needed to characterize the system in a satisfactory manner. This generates a notion central to genomic annotation: a notion of a “measurement space”. A measurement space, M , is comprised of a

collection of coordinate functions x_1-x_n over the set of interest. Objects such as ceiling positions or sequence fragments become imbedded in this space through the coordinate functions. Conversely, we can also say that the coordinate functions and the measurement space M parameterizes the objects, i.e. the sequence fragments. By this I mean that we can pick a point in the measurement space, say (GC%=50% and “TATAA is present”) and there is a corresponding set of sequence fragments whose coordinate function values are exactly (GC%=50% and “TATAA is present”).

Combining the notation in (1) and (2) crystallizes the structure of genomic annotation. We have the set S^G of all possible sequence fragments, to which we want to assign labels of biological information. Rather than do this directly, a set of coordinate functions is chosen to construct the measurement space. Next, a labeling function, $\phi_L: M \rightarrow L$ is constructed over the measurement space. Therefore, we have the three-way relationship:



The measurement space and coordinate functions introduce two key features to genomic annotation. First, the labeling action is made operational by the coordinate functions. Using the coordinate functions x_i allows us to use the operationally more convenient nominal definition of objects where one points to objects and says “this one is yellow, walks on four legs, produces milk, has hair,...and such objects are called dogs”. Second, the coordinate functions allow the utilization of the idea of a *space*. The idea of a space defines neighborhoods and distances which gives us the power to interpolate and extrapolate from existing knowledge. If we know that a particular point in our space has some property of interest, we might reasonably speculate that nearby points will also have a similar property (extrapolation); if we know that two different points have the same property, we might also speculate that points in between the two will also have the same property (interpolation).

The power of extrapolation and interpolation allows us to gain knowledge beyond existing knowledge. The goal of all gene detection, genomic annotation, and data mining methods, is to find a construction that best allows such inferential leaps. This goal will be best achieved if the objects form natural groups in the measurement space such that nearby objects share related biological properties. Foreshadowing my conclusions, I claim that the key to genomic annotation is to construct a set of measurements that become coordinate functions defining a measurement space, in which objects that share biological properties form natural groups.

Sequence-based and feature-based genomic annotation

In the next section I provide an overview of what I call sequence-based approaches and feature-based approaches. This distinction is mostly methodological and in many ways it is a poor distinction in terms of principles. In most genomic annotation problems, the only information we have is the sequence of the molecule and all inferences are sequence-based. In fact, in all of the methods we strongly rely on the assumption that the sequence is sufficient to define biology. In making this assumption we draw on two corollaries: first, that evolutionary descent relationships define the default sequence relationships and second, that the sequence-to-structure-to-biology relationship is supported well enough that we can use the sequence information to infer biological classes. As mentioned, in all of the methods, the basic idea is to create a “handle” on the sequences such that we can assign different sequences to different label classes. This process of creating handles involves constructing the right coordinate functions that parameterize the sequence fragments (i.e. the values of the coordinate functions index sequence fragment classes). Different kinds of parameterizations vary in the refinement of their values. For example, parameterizing individuals by their eye color is a coarse-grained parameterization while parameterizing by their retinal pattern is an extremely fine-grained parameterization. Too coarse-grained a parameterization yields an overly blunt tool for classification while too fine-grained a parameterization yields too inefficient a description (e.g. if every object has a unique value then nothing has been gained). The crux of the algorithms is to generate a parameterization that is flexible enough to yield both coarse-grained and fine-grained classification in the most efficient way possible.

Sequence-based approaches to genomic annotation are the more commonly used approaches. By a sequence-based approach I mean an algorithm or methodology that uses the linear sequence identity of the DNA or amino acid string as the critical input data. The simplest kind of method is a pattern search for a particular linear sequence, for example, the presence of an amino acid pattern [D-R-Y]. In the terminology I introduced above, a pattern search can be regarded as applying a coordinate function on a collection of sequence fragments that returns the value 1 when the pattern [D-R-Y] is present and returns the value 0 when such a pattern is absent. In this sense, every kind of sequence pattern such as those curated in the PROSITE database (Hoffman et al. 1999) or PRINTS database (Attwood et al. 1999) can be seen as a particular coordinate function. It is, of course, unusual for a particular sequence pattern to have biological implications itself. A particular pattern is usually seen as a signature of a particular class of molecules and the biological significance lies with the class label, say “MAP kinase”, rather than the particular sequence [R-E-R-D]. The use of a pattern can be regarded as a very coarse-grained extrapolation where all those sequences that share the pattern is considered to be in the

neighborhood of each other and share a molecular class (or a label) and all those that do not have the pattern have no neighboring relationship. The reason that such sequence patterns become reasonable signatures for a biologically meaningful class of molecules is that such classes are usually related to each other by common evolutionary descent – that is, the molecules are homologous. Homologous molecules often contain particular signature sequences as well as structural features that are conserved within the class and are useful indices of the class (Corpet et al. 1999; Lindahl and Elofsson 2000; Murzin et al. 1995; Srinivasarao et al. 1999). It is important to note that such sequence conservation is not the rule, but simply the consequence of a particular mode of evolution-neutral evolution, that has proven to be the pervasive evolutionary mode at the molecular level.

Since so-called homology modeling is an extremely important component of data-mining and annotation, it is worthwhile to review this notion further. These days it is common to equate sequence conservation with functional importance. However, this reasoning is not immediately obvious. For example, if the major mode of evolution is the accumulation of adaptive changes, the functionally most important parts of the molecule will be the least conserved. This can be easily seen with an engineering analogy. Suppose we were given a machine. If we were to try to engineer a more efficient version of the machine, logically the part that we would tinker with the most would be the functionally most important parts. Thus, the part that would change the least would be those parts that are functionally irrelevant or those parts that are constrained by physics (say, the need to reduce friction). On the other hand, if we were poor engineers, we would rarely make a change that made a functional difference and therefore we would only accumulate random irrelevant changes. (Of course, in both adaptive and neutral mode of change, detrimental changes would be quickly eliminated and they would not contribute to the observed changes.) There has been considerable debate as to whether natural evolution proceeds mostly through adaptive changes or through neutral changes (see Futuyma 1997). Accumulating molecular evidence since the 1970s has confirmed that, at least at the molecular level, the major mode of evolutionary change is neutral (Kimura 1983). Exceptions to this are the rare cases when in fact there is a burst of adaptive change. Such a burst of adaptive change often happens when the organism or the molecule enters a novel niche or acquires a novel function. [Such changes can become clues to molecular function (Gu 1999).] The high degree of divergence seen in *Drosophila* olfactory receptor genes that impeded their cloning (Clyne et al. 1999) might be due to such rapid adaptive changes – perhaps accompanying the radiation of arthropods.

As mentioned, taking a measurement that tests for the presence or absence of a particular linear sequence pattern is obviously very coarse. It would be desirable to gain efficiency and allow for contingencies through a more fine-grained extrapolation from a known sequence

pattern. This involves the extension of a particular pattern to a family of patterns. The simplest form of such an extension is to allow combinatorial deviations of the exact sequence. For example, we might consider not only the sequence [D-R-Y] but also [D-R-V] and [D-S-Y]. More generally, we might consider all possible combinations of a finite length fragment. For example, we might consider $20^3=8,000$ combinations of all possible fragments with a length of three amino acids. One might then construct a numeric function from the 8,000 different combinations such that high values of this function indicate some molecular class, say kinases. This numeric function – more often called the score function or the weight matrix, expands the family of patterns by giving the combinatorial variations a kind of continuity. That is, it allows us to know that [D-R-Y], [D-R-V], and [D-S-Y] are all similar to each other because they have all have a similar score, while [Y-L-C] is quite different because it has a very different score. This kind of construction can be seen as another level of functional construction. The set of possible fragments, S^G , is mapped by a set of combinatorial pattern coordinate functions to 0/1 values. In the three amino acid example, all sequence fragments are assigned presence/absence for each of the 8,000 possible combinations, resulting in an 8,000-dimensional 0/1 vector. The scoring function then maps the 8,000-dimensional vector to numeric values. This numeric value scale becomes our measurement space.

The length of the combinatorial variation modulates the grain of the generated measurement space. That is, a short fragment with a small number of possibilities generates a coarse-grained picture while a long fragment generates a very fine-grained picture. In practice, the length of fragments considered is of the order of tens of positions, such that a very large number of combinations are possible. For such cases, exact detailed construction of a score function will be impractical. Standard methods make an additivity assumption where a numerical value is given for the assignment of each residue at each position. Then the total score is computed as the sum of the values at each position (the so-called position specific score matrix; Bailey and Gribskov 1998; Bucher and Bairoch 1994; Gribskov et al. 1987). It is also common to give a probabilistic interpretation to such weights using sampling models of sequence subsets.

In principle, we could generate a sufficient set of coordinate functions with a comprehensive combinatorial variation including an allowance for insertions and deletions (Bucher and Bairoch 1994). However, such a construction can result in a very large collection of descriptors, thereby losing considerable efficiency. A key point in all of the methodological approaches is to generate an efficient handle on the S^G set. In principle, we could create a set of coordinate functions consisting of all possible three-dimensional folding structures of the fragments, but this would not be computable in practice. One way to gain efficiency is to replace combinatorial complexity with a smaller set of generators of combinatorial complexity. One family of such generators in-

volves using grammatical constructions (e.g. Dong and Searls 1994; Durbin et al. 1998; Grate et al. 1994; Krogh et al. 1994). Grammatical constructions model molecular sequences as if they were linguistic strings (for review see Searls 1997). For example, given a language, say American Standard English, certain strings of words (symbols) can be assessed (parsed) as belonging to the language or not belonging to the language (ungrammatical). Similarly, simple grammatical rules, and thus a language, can be designed such that a string of DNA or amino acids can be parsed as either belonging to the language or not belonging to the language. Those that belong to the language are considered members of a particular molecular class.

[In this usage, a language is a collection of concatenated symbols (strings) that is a subset of all possible such strings. Different languages are different subsets. Thus we might have strings comprised of the alphabet and punctuation. Of all possible such strings, only a particular subset comprises the English language and a different subset comprises the French language and so on.]

One of the simplest of grammatical constructions generating a language is the *regular grammar*. Regular grammar is a particular kind of grammar that allows generative rules of the kind $S \rightarrow aS$ (by which we mean that some word S can be replaced by S plus some symbol concatenated to the left). Different languages (and thus molecular classes) can be specified by designating various rules of this form. Regular grammar with some variations is commonly used in computer languages (e.g. PERL; regular expression) and the PROSITE database maintains a large categorization of molecular classes by regular expression languages. In the theory of computation, there are equivalences between grammatical constructions and models of computation. Regular grammar has an equivalent computational construct in finite state automata (Davis et al. 1994; Durbin et al. 1998).

Recall that the whole idea is to extend or extrapolate a given sequence pattern into a larger family of patterns. Regular expressions allow a handle into a very large family of patterns with quite a bit of flexibility. The restriction is that the specification of a particular family can be only done in a linear fashion because the rules are of the form $S \rightarrow aS$. That is, we can only specify a molecule by saying A should follow B, followed by C or D, etc. More complicated specifications can be generated using more complex grammar. For example, context free grammar which allows rules of the form $S \rightarrow aSb$ has been used to model RNA fold families (Brown and Wilson 1996; Grate et al. 1994). The rule of the game is how to generate the most efficient set of “handles” into the complexity of biologically relevant molecular classes. A grammatical specification is efficient since a particularly small set of symbols, say $S \rightarrow aS$, can represent possibly infinite set of strings (for this example, all strings of the form $a, aa, aaa,$ and so on).

Returning to the central problem, the goal of genomic annotation is to assign labels to subsets of genomic sequence fragments, S^G . A cartoon of this process is shown

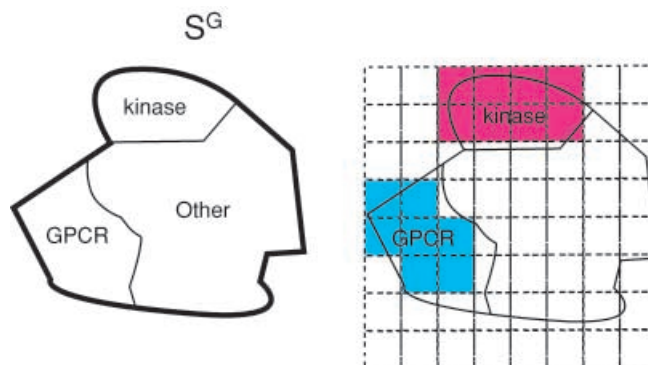


Fig. 1 A cartoon of the genomic annotation process. All possible sequence fragments of a genome are shown as a region denoted S^G . Different annotation classes such as protein classes form a partition of S^G . Since S^G is a complex object, often a set of “handles” such a presence/absence of a particular motif is used to pre-divide S^G . This is shown *on the right* as a grid. The pre-divided pieces are then used to annotate the sequences. If the divisions are not compatible with the real sequence classes, errors are induced in the annotation (shown as *shaded blocks*)

in Fig. 1. Conversely, we can think of the labeling function as “cutting up” the S^G set into different categories – that is, categorizing S^G into biological classes. Different annotation functions need to generate different kinds of cuts according to the particular molecular class that is being delineated. S^G is an extremely large collection of sequence fragments and we cannot work on them directly. Sequence-based methods use a set of handles based on the sequence pattern to divide up S^G into more manageable pieces – that is, the handles generate a pre-categorization of the sequences. Annotation is then done on the pre-divided pieces (Fig. 1). For example, all of S^G might be divided into those strings that start with the base A, C, G, or T. Then we might say, those that start with A and C are kinases. Ideally, the pieces generated by the handles should be compatible with the cuts needed by the annotation function. The previous example would be a very bad handle. A combinatorial handle utilizing more sequence positions can divide S^G very finely, thus it has a better chance of matching the annotation needs. However, in the process we generate a large number of handles that are again difficult to manage. A generative grammar based handle allows the generation of a flexible set of pieces with a small number of handles. By “flexible”, I mean that it can be used to specify “complex cuts” of S^G (which might be needed for complex classes of annotation) without generating a lot of complexities of the handles themselves.

Despite their flexibility, generative grammars are still crude tools for dividing up S^G . Similar to position specific weight matrices, we can extend the flexibility of how the grammars “cut” S^G by numerically parameterizing the grammars using a stochastic version. So rather than having a fixed rule $S \rightarrow aS$, we can have $S \rightarrow aS$ with probability 0.4 and $S \rightarrow \beta S$ with probability 0.6. Given a sequence string, instead of strictly assessing whether it belongs in the language, we can attach a numerical value

that is interpreted as the probability of belonging to the language. As mentioned, grammars have equivalent constructs in terms of finite state automata. In the case of regular grammars, it is often more convenient to use the automata representation to specify the stochastic parameterizations since it also can be made to resemble finite state Markov models (Durbin et al. 1998). The Markov models lend themselves to a more natural stochastic process intuition. (Although the usual left-to-right specification is a bit awkward since there is no actual generative process of sequences in that manner. The Markov structure is better understood as defining dependence relationships between sequence positions, rather than as a temporal process of generating sequences.) In particular, a very convenient set of handles can be created by incorporating a set of states that are hidden from direct observation (Hidden Markov models; Brown et al. 1993; Churchill 1989; Krogh et al. 1994). It is common to assume that the hidden states represent a kind of model-theoretic structure of the observed sequence string, such as the structural properties of the protein or some systemic property of the genome (e.g. CG islands; Durbin et al. 1998). One of the more interesting idea with many applications is the construction of a very general Hidden Markov model whose parameter sets (i.e. the transition probabilities) become indices into different proteins families (see Eddy 1998). This general model is like an all-purpose knife whose cuts change upon different parameter settings.

Feature-based annotation methods are relatively recent. The idea directly follows the basic construction I outlined at the beginning. First, a set of coordinate functions is constructed from the S^G set to form the measurement space (also called a feature space). Coordinate functions in these approaches are usually more straightforward. For example, it could be measurements like GC content, some kind of hydrophathy index (e.g. Engelman et al. 1986; Kyte and Doolittle 1982), codon usage (e.g. Powell and Moriyama 1997), amino acid frequencies (Wootton 1994), physical properties (Baldi et al. 1998), and so on. In our work on *Drosophila* olfactory receptor proteins, we used statistical descriptors of quasi-periodic changes in hydrophathy, as well as a weighted index of amino acid frequencies (Clyne et al. 1999; Kim et al. 2000). One key advantage in this construction is that because all sequences are placed in an explicitly constructed continuous space, we have considerable flexibility in delineating the annotation regions. That is, the cuts can be specified very flexibly using explicit geometric techniques. For example, a numerical function can be constructed over the measurement space whose contours delineate different annotation groups (Fig. 2a). This is one example of a discriminant function (McLachlan 1992) the idea of which is to classify points in a space (the measurement space) by a geometric description of boundaries (for applications see Chou and Elrod 1998; Kihara et al. 1998; Kim et al. 2000). Using explicit geometry can yield considerable advantages in that we have a natural interpretation of ideas of interpolation and

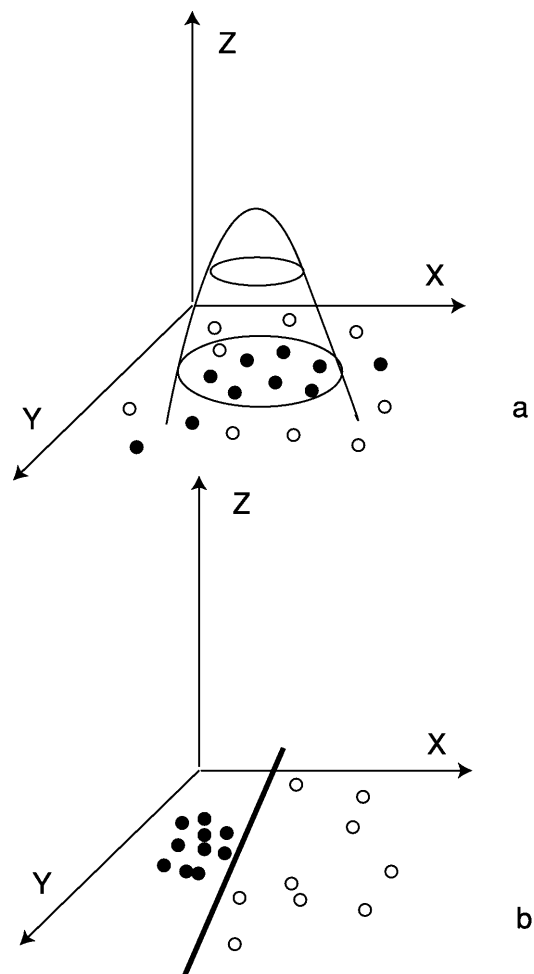


Fig. 2a, b A schematic diagram of a discriminant function for annotating genes. The x and y axes represent hypothetical measurements that give coordinates to sequence fragments (see main text). The *filled black circles* represent an annotation class, say kinases, and the *open circles* represent other miscellaneous sequences. The z axis is a function whose values discriminate the different annotation classes. **a** A quadratic discriminant function is shown as a parabolic curve. The ellipse in the middle is the contour where the quadratic function has positive values. The ellipse “captures” as much of the black circles as possible. **b** The ideal case in which the measurement functions induce the annotation class (*black circles*) to form a compact geometric object. In this case, a simple line delineates the annotation class

extrapolation. Ideal conditions for constructing the most effective annotation would be if all sequence fragments that belong to some molecular class occupy a compact region in the measurement space (Fig. 2b). In this case, we only need to draw a line and annotate everything on one side as, say, “G protein-coupled receptors”, and the other side as “not”.

The crux of getting all sequence fragments of a particular class to be positioned in a compact manner is in the construction of the measurement space. If we measure the “right” things, natural groups, such as proteins of a particular function, should appear contiguous. The measurement space described up to now has been explicit-

it in the sense that we were talking about actual measured quantities, such as the GC content of a sequence. As mentioned, these measured properties assign coordinate values for each sequence. A more abstract notion of a space can be defined without an explicit set of coordinates by defining a notion of distance among objects. A simple example would be to take every pair of sequences, align the pair, and compute their percentage mismatch. An idea called Support Vector Machines (Burges 1998) combines a flexible notion of distances between pairs of objects with the idea of obtaining compact natural groups. Jaakkola et al. (1999) have applied this idea to gene identification by defining a distance between sequences using Hidden Markov model parameters.

Summarizing the discussion up to this point, the problem of genome annotation is to assign biologically informative labels to a set of sequence fragments. We do not have a constructive way of assigning biologically informative labels except in very rare cases. Rather, we have a few case study examples where the labels are known from empirical experiments. The goal of the annotation methods is to use this prior knowledge in the most efficient manner possible. Thus, the ideas of extrapolation and interpolation become crucially important. The prior knowledge consists of particular exemplars, and in order to extrapolate or interpolate from the exemplars, we need “handles” to reach to other sequence fragments. The most natural handle is an embedding of the exemplars in a space and using the way we normally think of extrapolating and interpolating points. This space is the measurement space. Constructing an optimal measurement space underlies the complexities of all the different approaches such as motifs, profiles, Hidden Markov models, Fisher kernels, and so on. The sequences are an extremely large collection of objects and the handles reduce this complexity by mapping a large number of sequences to a small number of handles. For example, a particular sequence motif such as TATAAT is a single handle that maps a large number of sequences. Or, the statement “all sequences that fall within a 10% similarity radius of sequence X” is another simple handle that maps a large number of sequences. But, as such handles are constructed a frustrating thing happens. More often than not, we have the need to delineate (“get a handle on”) a complex set of sequences. This requires the creation of an increasingly flexible set of handles by incorporating such things as weights, stochastic parameters, etc. As these extensions are incorporated, the handles themselves become too complex. Other tools such as grammatical constructions are brought in to manage the problem of handle complexity but there is a vague feeling that this frustration is inherent to the problem.

At an abstract level, genomic annotation is the operation of representing one set of objects, the sequences, with another set of objects, biological categories. If there is a particular degree of inherent and meaningful complexity to the first set there doesn't seem to be any way to reduce the complexity of the second set without a loss of meaning. Therefore, the whole concept of genomic

annotation depends on there being an inherent and meaningful *order* to the sequence set that maps to the biological information. At some levels of description, this is obvious; for example, a sequence feature like the TATA box. At other levels of description, especially teleological descriptions like “a gene for eyes”, things become more murky. Ideally, the most efficient annotation method should reveal the most natural structure of the sequence data that is the most informative basis of biological inference. Generating a classification (annotation) that is most informative, predictive, and efficient is the main goal of biological systematics and classification (Sneath and Sokal 1973; Stevens 1994). Thus, it is with no surprise that tools that have been used in systematics such as evolutionary trees, discriminant functions, and geometric projections, are directly relevant to the problems of genomic annotation. Like problems of systematics, the genomic annotation problem is essentially an inductive problem of revealing the groupings of natural structures. And, like all inductive problems, there is no algorithm or theory to define the best methodology.

Conclusions

I have outlined above the broad theoretical structure for genomic feature annotation. Hundreds if not thousands of programs have been created for genomic feature annotation and have been applied in genome projects. However, many instances of applying computational approaches are not driven by a particular biological problem. For example, the *Drosophila* annotation Jamboree (Anonymous 2000) was a highly interesting application of the collective knowledge of the *Drosophila* community, but was not driven by an immediate need to search for, say, olfactory genes to further understand the organism. This creates a certain sense of “generic knowledge” that might not be particularly useful to the individual researcher. “Uncommitted” annotation is also far more likely to propagate annotation errors (Brenner 1999). (On the other hand, from this point, the *Drosophila* Jamboree was far ahead in that it involved researchers with actual research programs in the particular molecule class. That is, it involved “committed” individuals to whom the results really mattered.)

Gene identification driven by a particular biological research project generates an ideal combination of circumstances for genomic annotation. But it also generates several practical problems. One problem that often arises is that genome projects pay close attention to the analyses and presentation of the data, but often the raw data needed to carry out specific projects is under-curated. Therefore, the web pages associated with the genome projects have many interesting features until one gets to the FTP page to download the raw data, which are often scattered in cryptically named files. But foremost of the practical problems is that particular projects have particular needs that are not solved by a generalized tool. For example, one might want to search for novel G protein-

coupled receptors with a particular set of upstream binding sites. While the specification is simple, the details become problematic (e.g. how many binding sites within how many bases) and surprisingly cumbersome for the generic user who may have to use many different available tools and examine megabytes of output data. Thus, while a large set of tools is available, especially through the web-based application servers, their utility can be limited for even a well-versed user. A laboratory manual provides an excellent set of tools for laboratory work, yet the manual itself would be insufficient for actual projects. Another “problem” is that many of the practical problems are not solved so much by computational or algorithmic advances but rather by careful modeling of the biological problem. Many of the advances in gene prediction came from continued incorporation of biological data modeling (e.g. Krogh 2000; Kulp et al. 1997; Lukashin and Borodovsky 1998). This may seem obvious, but I believe the best bioinformatics tool for genomic annotation is not a computational method, but an individual with training in biological data modeling and the ability to implement a set of customized computational tools.

The gap in our knowledge between sequence data and biological information may never be filled. Still, computational biology and bioinformatics remain one of the critical tools for filling this gap. But much of the nature of work is inductive inference rather than deductive reasoning. No algorithm or theory exists to offer the best inductive tool for any given inferential problem. As in many things biological, a keen insight into inductive problems relies more on intuition and imagination than mathematics or computation. Of course, mathematics and computation are critically important in determining what is possible. Determining what is beyond the possible is the task of biological informatics.

References

- Agarwal P, States DJ (1998) Comparative accuracy of methods for protein sequence similarity search. *Bioinformatics* 14:40–47
- Anonymous (2000) *Science* 287, no. 5461
- Attwood TK, Flower DR, Lewis AP, Mabey JE, Morgan SR, Scordis P, Selley JN, Wright W (1999) PRINTS prepares for the new millennium. *Nucleic Acids Res* 27:220–225
- Bailey TL, Gribskov M (1998) Methods and statistics for combining motif match scores. *J Comp Biol* 5:211–221
- Baldi P, Chauvin Y, Brunak S, Gorodkin J, Pedersen AG (1998) Computational applications of DNA structural scales. *Intelligent Syst Mol Biol* 6:35–42
- Boguski MS (1999) Biosequence exegesis. *Science* 286:453–455
- Bork P (2000) Powers and pitfalls in sequence analysis: the 70% hurdle. *Genome Res* 10:398–400
- Brenner SE (1999) Errors in genome annotation. *Trends Genet* 15:132–133
- Brown M, Wilson C (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. *Pac Symp Biocomp* 1:109–125
- Brown M, Hughey R, Krogh A, Mian IS, Sjolander K, Haussler D (1993) Using Dirichlet mixture priors to derive hidden Markov models for protein families. *Intelligent Syst Mol Biol* 1:47–55
- Bucher P, Bairoch A (1994) A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. *Intelligent Syst Mol Biol* 2:53–61
- Burges CJC (1998) A tutorial on Support Vector Machines for pattern recognition. *Data Mining Knowledge Discovery* 2:121–167
- Chou KC, Elrod DW (1998) Using discriminant function for prediction of subcellular location of prokaryotic proteins. *Biochem Biophys Res Commun* 252:63–68
- Churchill GA (1989) Stochastic models for heterogeneous DNA sequences. *Bull Math Biol* 51:79–94
- Clyne PJ, Warr CG, Freeman MR, Lessing D, Kim J, Carlson JR (1999) A novel family of seven transmembrane proteins are candidate odorant receptors in *Drosophila*. *Neuron* 22:327–338
- Corpet F, Gouzy J, Kahn D (1999) Recent improvements of the ProDom database of proteins domain families. *Nucleic Acids Res* 27:263–267
- Davis MD, Sigal R, Weyuker EJ (1994) Computability, complexity, and languages: fundamentals of theoretical computer science. Academic Press, New York
- Dong S, Searls DB (1994) Gene structure prediction by linguistic methods. *Genomics* 23:540–551
- Durbin R, Eddy SR, Krogh A, Mitchison GJ (1998) *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, New York
- Eddy SR (1998) Profile hidden Markov models. *Bioinformatics* 14:755–763
- Engelman DM, Steitz TA, Goldman A (1986) Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. *Annu Rev Biophys Biophys Chem* 15:321–353
- Fickett JW (1996) Finding genes by computer: the state of the art. *Trends Genet* 12:316–320
- Fickett JW, Wasserman WW (2000) Discovery and modeling of transcriptional regulatory regions. *Curr Opin Biotechnol* 11:19–24
- Futuyma DJ (1997) *Evolutionary biology*, 3rd edn. Sinauer, Sunderland, Mass.
- Grate L, Herbster M, Hughey R, Haussler D, Mian IS, Noller H (1994) RNA modeling using Gibbs sampling and stochastic context free grammars. *Intelligent Syst Mol Biol* 2:138–146
- Gribskov M, McLachlan AD, Eisenberg D (1987) Profile analysis: detection of distantly related proteins. *Proc Natl Acad Sci USA* 84:4355–4358
- Gu X (1999) Statistical methods for testing functional divergence after gene duplication. *Mol Biol Evol* 16:1664–1674
- Hoffman K, Bucher P, Falquet L, Bairoch A (1999) The PROSITE database: its status in 1999. *Nucleic Acids Res* 27:215–219
- Jaakkola T, Diekhans M, Haussler D (1999) Using the Fisher kernel method to detect remote protein homologies. *Intelligent Syst Mol Biol* 4:149–158
- Kihara D, Shimizu T, Kanehisa M (1998) Prediction of membrane proteins based on classification of transmembrane segments. *Protein Eng* 11:961–970
- Kim J, Moriyama EN, Warr CG, Clyne PJ, Carlson JR (2000) Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. *Bioinformatics* (in press)
- Kimura M (1983) *The neutral theory of molecular evolution*. Cambridge University Press, New York
- Krogh A (2000) Using database matches with HMMGene for automated gene detection in *Drosophila*. *Genome Res* 10:523–528
- Krogh A, Brown M, Mian IS, Sjolander K, Haussler D (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol* 235:1501–1531
- Kulp D, Haussler D, Reese MG, Eeckman FH (1997) Integrating database homology in a probabilistic gene structure model. *Pac Symp Biocomp* 2:232–244
- Kyte J, Doolittle RF (1982) A simple method for displaying the hydrophobic character of a protein. *J Mol Biol* 157:105–132
- Lindahl E, Elofsson A (2000) Identification of related proteins on family, superfamily, and fold level. *J Mol Biol* 295:613–625

- Lukashin AV, Borodovsky M (1998) GeneMark.hmm:new solutions for gene finding. *Nucleic Acids Res* 26:1107–1115
- McLachlan RGJ (1992) *Discriminant analysis and statistical pattern recognition*. Wiley, New York
- Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP:a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 247:536–540
- Powell JR, Moriyama EN (1997) Evolution of codon usage bias in *Drosophila*. *Proc Natl Acad Sci USA* 94:7784–7790
- Reese MG, Kulp D, Tammana H, Haussler D (2000) Genie-gene finding in *Drosophila melanogaster*. *Genome Res* 10:529–538
- Searls DB (1997) Linguistic approaches to biological sequences. *Comput Appl Biosci* 13:333–344
- Singh GB (2000) Computational approaches for gene identification. *Methods Mol Biol* 132:351–364
- Sneath PHA, Sokal RR (1973) *Numerical taxonomy*. Freeman, San Francisco, Calif.
- Srinivasarao GY, Yeh LS, Marzec CR, Orcutt BC, Barker WC (1999) PIR-ALN: a database of protein sequence alignments. *Bioinformatics* 15:382–390
- Stevens PF (1994) *The development of biological systematics: Antoine-Laurent de Jussien, nature, and the natural system*. Columbia University Press, New York
- Wootton JC (1994) Sequences with “unusual” amino acid compositions. *Curr Biol* 4:413–421