Integrated graphical analysis of protein sequence features predicted from sequence composition

Erik L.L. Sonnhammer[1,2] and John C. Wootton[2]

[1]Center for Genomics Research
Karolinska Institutet
171 77 Stockholm, Sweden

[2]Computational Biology Branch,
National Center for Biotechnology Information,
National Library of Medicine, Building 38A, Room 8N805
National Institutes of Health,
Bethesda, Maryland 20894, USA

Email:
Erik.Sonnhammer@cgr.ki.se,
wootton@ncbi.nlm.nih.gov

Running title: Integrated graphical sequence analysis

Keywords: sequence analysis, graphical visualization, dot-plot, database search viewing, sequence complexity, transmembrane, coiled-coil, protein structure, non-globular proteins, algorithms, data definition format

ABSTRACT

Several protein sequence analysis algorithms are based on properties of amino acid composition and repetitiveness. These include methods for prediction of secondary structure elements, coiled-coils, transmembrane segments or signal peptides, and for assignment of low-complexity, non-globular, or intrinsically unstructured regions. The quality of such analyses can be greatly enhanced by graphical software tools that present predicted sequence features together in context, and allow judgement to be focused simultaneously on several different types of supporting information. For these purposes, we describe the SFINX package, which allows many different sets of segmental or continuous-curve sequence feature data, generated by individual external programs, to be viewed in combination alongside a sequence dot-plot or a multiple alignment of database matches. The implementation is currently based on extensions to the graphical viewers Dotter and Blixem, and scripts that convert data from external programs to a simple generic data definition format called SFS. We describe applications in which dot-plots and flanking database matches provide valuable contextual information for analyses based on compositional and repetitive sequence features. The system is also useful for comparing results from algorithms run with a range of parameters to determine appropriate values for defaults or cutoffs for large-scale genomic analyses.

INTRODUCTION

Any protein sequence, as typically inferred from a genomic or mRNA sequence, potentially represents a rich mosaic of molecular properties reflecting structure, dynamics, interactions and roles in cellular machinery. Interpretation and annotation of such a sequence is a complex conceptual task, which is usually achieved by a synthesis of algorithmic analysis and expert judgement. Individual algorithms vary in their ability to diagnose or classify various sequence features, and knowledgeable human interpretation is generally considered o be essential. Even seemingly straightforward outputs, such as database sequence similarity search results using conservative cutoffs, are frequently greatly enriched by human abilities to perceive context, associations and unexpected pitfalls. In all cases, graphical display can dramatically improve envisioning and comprehension of the interrelated sets of data, and most sequence analysis software packages include graphical tools.

In addition to comparative analysis of conserved domains and sequence motifs by means of database searches, several algorithms have been designed to predict certain protein features primarily from attributes of composition and repetitiveness. Such features include secondary structure elements, transmembrane segments, signal peptides, low-complexity regions, coiled-coils, other non-globular domains, and intrinsically unstructured regions. These results are typically interpreted, together with regions of sequence conservation, to infer a provisional map of the possible structural and functional regions of a protein. This task presents several difficulties and requires critical evaluation of results from various compositional, alignment, and modeling algorithms.

To assist these tasks, adaptable software is needed that takes the results of different amino acid sequence feature analysis programs and uses them as inputs into graphics programs designed for integrated visualization. Also needed is the ability to run each program with different parameter sets and compare the results graphically. Weighing the significance of

different types and levels of evidence together usually leads to a more accurate analysis than running each prediction program separately with default parameters. In addition, integrated analyses of this type are valuable in calibrating parameters during development of computational methods, for example to employ them in large-scale genomic analysis. Many analysis programs are provided with very permissive default parameters to minimize false negatives, whereas in genome-wide analysis it is often important to use non-default conservative parameters to limit the number of false positives.

It is desirable, therefore, to view the combined output from several approaches, algorithms, and parameter sets, in many cases juxtaposed with database matches. Here, we describe a flexible software system that meets these various needs, and illustrate some of its applications. Since it is impossible to define exact rules on how to interpret such multi-facetted data, we provide a set of typical examples that illustrate how logical reasoning based on the combined output of many different analyses can lead to a correct interpretation, or at least avoidance of an incorrect one.

Data types and formats

There are in principle two primary types of data for describing sequence features: s*egments* and *curves*. Segments are defined by one start and end sequence coordinate. Typically, the sequence between these coordinates is assigned a certain property algorithmically, such as a low complexity region. Curves (or "profiles"), in contrast, consist of an array of scores, each score being assigned by an algorithm to a single residue. We here use the term "curve" because the term "profile" is mainly used in sequence analysis to denote a matrix of numbers along the sequence. Segments frequently have a score too, and may have associations with other pieces of data, particularly if they are "matching segments" that can be aligned by similarity to other sequences or sequence models. It is often advantageous to browse matching segments from database searches at the level of aligned residues; a special viewer for this purpose is Blixem (Sonnhammer and Durbin 1994).

Data sets, of both segment and curve types, can be obtained either by parsing the output of available sequence analysis programs or by independent calculation from the sequence being analyzed. Many prediction programs not only produce a set of segments as output, but also calculate a profile internally, according to some mathematical function or empirical scale, as part of the algorithm. This is the case in, for instance, the SEG complexity analysis (Wootton and Federhen 1993; Wootton and Federhen 1996), most transmembrane segment prediction programs and secondary-structure prediction methods. Generally, in these cases, the underlying profile may be readily calculated by using the appropriate function, independently of the program. Some programs report both the segments and the underlying profile, for instance COILS2 (Lupas *et al.* 1991), that predicts alpha-helical coiled-coils.

A number of established database and visualization systems exist that include built-in functions for sequence segment display. These include ChromoScope (Zhang *et al.* 1994), bioWidgets (Searls 1995), APIC (Bisson and Garreau 1995), the BDGP java sequence viewer (Rubin 1996), GAIA (Bailey *et al.* 1998), and ACEDB (Durbin and Thierry-Mieg 1999). These are relatively large software suites that require a significant investment in knowledge to become operational, usually due to the intricacies of specifying a practical data model. For instance, the data definition languages such as ACEDB and ASN.1 were designed to store biological objects in a rigorous way. Generating and parsing data in such formats involves supporting a substantial framework of semantic rules. For data consisting only of segments or

curves, the complications of conforming to such a format are unwarranted, and a simple tabular format is adequate. Furthermore, many of the available visualization systems have various limitations, depending on their history of development, which in many cases was oriented towards displaying genetic or physical maps, and thus have no facility for curve data. To our knowledge, only the commercial APIC system was designed to handle curve data in a generic way.

In contrast to these large, comprehensive systems, our goal is to provide simple, yet powerful, generic tools that allow any sequence crunching program to communicate its results to any graphical viewer. At the core is a simple data format for sequence feature series, which we call SFS. Sequence analysis programs typically produce data that is compatible with the present SFS data model, but it is also extensible to incorporate features that may need special treatment in the future. SFS achieves a logical separation of prediction/calculation programs and viewers, and thus removes the need for special visualization tools for each individual program. Viewers can then become more powerful and evolved tools, while the algorithmic implementations can be developed without the extra burden of building visualization tools. The overhead for both viewers and calculation programs to support the lightweight SFS format is minimal.

The two core data types in the SFS format are segments and XY curves. An XY curve is a two-dimensional plot of a series of X and Y value pairs, where X is the sequence residue coordinate. The information stored is very reduced, but is sufficient for generating a rich and easily interpretable graphical representation. In addition to the coordinates and score, each data point is associated with information necessary to link data points from a common source together and a color to distinguish it graphically. Optional annotation is allowed. However, the precise shape or placement on the screen of an object can not be stored explicitly; this is a property of each particular viewer, and only generic attributes can be specified in SFS. This follows the idea behind the HTML markup language. The SFS format is likewise intended to work with browsers via the World Wide Web, using SFS-viewing helper applications.

Recently, two systems for sequence feature markup have been described that are based on XML, which is an extension of HTML: BIOML (Fenyo 1999) and BSML (Spitzner 1999). XML is a structured format for data exchange that is becoming increasingly popular, particularly for describing data objects of hierarchical nature. However, because of the flexibility of XML to describe in principle any data with any syntax and semantics, writing an XML parser is far from trivial. We do not consider typical sequence features complex enough to motivate the complexity of generating and parsing XML. The main motivation for inventing SFS was to keep the format so simple that it becomes almost trivial to generate and parse the data, yet powerful enough to describe all typical types of features. In principle, an XML block corresponds to a field in SFS, hence converting SFS to XML and vice versa is straightforward. Hierarchical levels are not usually used for describing sequence features, but multiple attributes may be, e.g. the color and shape of a feature. The tabular SFS solves this by concatenating multiple attributes in a comma delimited list in a single field. Because XML has gained popularity in the bioinformatics community, we provide a tool for conversion of SFS to XML, and allow the results on the WWW server to be returned in XML.

A simple data format similar to SFS also exists in ACEDB for importing ‘user segments’ into the sequence map display. Another format used to exchange data between a number of gene prediction groups is the GFF format for gene-finding features (http://www.sanger.ac.uk/Software/GFF/), which is now also supported by ACEDB. Both

these formats support one single data type for sequence segments. Since GFF is essentially a simpler version of SFS, it is also supported directly by the viewers presented here.

We describe here two graphical viewers that support the SFS format and integrate segment and curve features into their rather specialized graphical analysis: the Dotter dot-plot program and the Blixem database-search results viewer. Previous versions of both these programs had some rudimentary displays of segmental features, but they have now been upgraded to accommodate any number of SFS data series.

Dotter (Sonnhammer and Durbin 1995) is a full dot-plot calculation program which stores the score of each cell in a dot-matrix. The stringency of the dot-plot analysis can be set interactively, using Dotter's dynamic "Greyramp" tool during viewing of the plot, without having to recalculate the dot-matrix. Displaying sequence features calculated by other programs together with a self-dot-plot is particularly useful for analyzing internal repeats and regions of compositional similarity. Similarly, Dotter can be used to analyze whether features of two different sequences make sense in the context of the similarity provided by a dot-plot. The size of the "sliding window" used to generate the dot-plot is by default set to the expected length of a high-scoring segment pair in Dotter, but can also be set manually to focus on repeats of a certain periodicity. It is often useful to explore the dot-plot with different window sizes. Potentially, a window size of 1, showing all similarities at the single-residue level, contains the maximum compositional information content, but this tends to obscure diagonals corresponding to repeated motifs.

Blixem (Sonnhammer and Durbin 1994) shows database matches generated in a BLAST search in a slave-master alignment. It is valuable to combine sequence features, which may, for example, suggest domain boundaries or functional characteristics, together with the database matches, thus achieving a more accurate interpretation. Blixem has two panels; the top panel shows a schematic overview of features and database matches along the entire query sequence or in a zoomed in region. A sliding box in the overview panel frames a region that is displayed in the bottom panel, in which features and database matches are shown in colored residue letters. Blixem can also be used without showing BLAST matches, in which case it simply acts as a general graphical data viewer for any sequence feature.

We focus here on applications of the SFS format for detailed analysis of compositional and repetitive protein sequence features, and for parameter calibration, employing readily available calculation and prediction programs. For these particular programs, we provide user-friendly scripts to run them, convert the output to SFS, calculate various profile curves, and to view the combined output in Dotter and Blixem. The entire package of scripts, parameter sets, and viewers is called SFINX. The scripts dotOmni and blxOmni run all incorporated analyses and present the results in a viewer as a single action. Additional analysis programs can be incorporated into the system with little effort.

RESULTS

In this section, we demonstrate particular applications of the SFINX package to analyses of compositionally biased and repetitive regions, transmembrane segments, and alpha-helical coiled-coils in amino acid sequences. The role of graphical visualization needs to be understood in the context of the underlying theories, goals, and evaluation criteria of each of these methods.

Compositional complexity and repeat analysis

Many regions of contrasting compositional bias occur in both nucleotide and amino acid sequences (Karlin and Brendel 1992; Salomon and Konopka 1992; Wootton 1994a; Wootton and Federhen 1993). Investigation of local compositional complexity and periodicity is informative at an early stage of the analysis of a new protein sequence, particularly when results can be interpreted together with local matches from database searches (Altschul *et al.* 1994; Wootton and Federhen 1996). In natural protein sequences, there is a strong tendency for compact globular folded domains to have a high complexity of composition that resembles a "random" distribution of amino acid frequencies (Wootton 1994a; Wootton 1994b). In contrast, compositionally biased regions of lower complexity correlate in most cases with non-globular, extended or intrinsically unstructured regions (Dunker *et al.* 1998; Wootton 1994b; Wright and Dyson 1999). Numerous low complexity protein regions are involved in crucial molecular functions and interactions, but, in general, they are relatively intractable to structural investigation by crystallographic methods, in contrast to globular domains (Wootton 1994a). Increasingly, NMR methods are yielding information on the dynamics and interactions of conformationally flexible low-complexity domains (Wright and Dyson 1999).

Compositional complexity analysis provides, therefore, a general method for investigating architectural features of polypeptides, especially for making provisional assignments of some domain boundaries in multi-domain proteins (Wootton and Federhen 1996). Simple complexity measures and segmentation algorithms have been described previously (SEG, PSEG for protein sequences, NSEG for nucleotide sequences (Wootton 1994a; Wootton and Federhen 1993). These identify optimal segments of low complexity, subject to parameters ("window length", "trigger complexity", and "extension complexity") that control the stringency and granularity of the analysis. Relatively long windows, for example 45 residues, are often appropriate when SEG is used in searches for long, potentially non-globular regions of proteins (Wootton 1994b; Wootton and Federhen 1996). However, a much more comprehensive analysis is achieved by using a range of parameter values and by integrated visualization of several measures of sequence complexity. Complexity profiles, calculated at different sliding window lengths, and self-similarity dot-plots also provide useful visual checks on the actual data underlying the algorithmically assigned segments.

Low complexity segments may have approximate or exact sequence repeats or may lack regular or recurrent patterns. The attribute of regular periodicity can be analyzed independently of overall compositional complexity, by calculating the sequence complexity only for residues that are spaced at a defined interval from each other. This is implemented in the SFINX package using the PSEG program described previously (Wootton and Federhen 1996).

A complementary approach, named HISEG, is also implemented in the package. This variant of the SEG algorithm reports optimized sequence segments of high, rather than low, complexity. HISEG segments have the greatest local compositional complexity (or greatest “randomness”) based on a uniform distribution, or any arbitrarily specified distribution, of amino acid frequencies, subject to the same stringency and granularity parameters as SEG. In practice, HISEG is less precise than SEG for definition of the boundaries between adjacent regions of contrasting complexity, because optimal matches to the target frequencies tend to extend beyond high-complexity segments into more biased regions (Wootton and Federhen, unpublished). Consequently the segments predicted by HISEG often overlap those assigned by SEG and the latter usually more accurately indicate the appropriate boundaries.

Nevertheless, the complementary properties of HISEG and SEG are valuable when the results of both methods are viewed together, because their predictions tend to correspond approximately to, respectively, globular and non-globular domains, as illustrated below.

To accommodate the different types of compositional complexity we run SEG and display entropy curves with 4 different window sizes: 12, 25, 45, and 75. For each window size, SEG is run with three empirically selected parameters for "stringent", "medium", and "relaxed" modes. For stringent mode, we used trigger and extension cutoffs of (2.0, 2.3), (2.95, 3.25), (3.3, 3.65) and (3.55, 3.75) for the different window sizes. For medium mode we used (2.2, 2.5), (3.0, 3.3), (3.4, 3.75), and (3.65 3.85), while for relaxed mode (2.35, 2.65), (3.15, 3.45), (3.5, 3.8), (3.7, 3.95). PSEG is run with periodicities 2 through 12 with trigger and extension complexity cutoffs set to 1.5. These cutoffs were set empirically in order to mainly reveal low complexity segments of significance.

Coiled-coil analysis

A particular form of repetitive protein sequence is the heptad repeat found in most alpha-helical coiled-coil proteins. These coils can consist of either two or three helices wound around each other in an extended rod-like structure. Lupas *et al.* (1991) developed a general prediction method for predicting coiled-coil subsequences, based on the position-specific biases within the heptads. In the present implementation, COILS2, (Lupas 1996), predictions can be run using two scoring matrices, "MTI" and "MTIDK", which are based on different sets of examples. One may also vary the window length, and run it with or without position-specific weighting. It is generally inadequate to use only a single combination of these parameters, because false-positive predictions tend to occur with some of them. However, a good judgement of the appropriate balance between sensitivity and specificity can be achieved using 12 combinations of these options and comparing the results graphically, as implemented in the SFINX package. These 12 parameter combinations are obtained by using each of the 4 possible combinations of matrix and weights, with the three window sizes 14, 21, and 28.

Combined complexity and coiled-coil analysis

Sequences encoding coiled coils always contain regions of low sequence complexity and short repetitions. The common type of coiled coil with a heptad repeat, the target of COILS2, is normally associated with low complexity segments reported by SEG with the above parameters. In addition, most such sequences give a PSEG segment in period 7 only, but this is not always the case because many different types of coiled coils exist. Of 272 sequences from SWISS-PROT 39 (Bairoch and Apweiler 2000) with annotated coiled coils of length 100 or more, 57% (154) produced a PSEG period 7 segment. Globular proteins generally do not produce such segments. In PDB (Sussman *et al.* 1998), which consists of mainly globular proteins, 1 % (128 of 8997) of the sequences produced a PSEG period 7 segment. It should be noted that alternative types of coiled coils, e.g. the triplet type found in collagen, is not detected by COILS2, but is readily detected by PSEG.

Figure 1 illustrates the value of combined interpretation using the complementary approaches of complexity and coiled-coil predictions with different parameters. In this example, the N-terminal non-globular region of *T. thermophilus* seryl tRNA synthetase is known from a high-resolution crystal structure determination to be mostly an extended, antiparallel, two-stranded coiled-coil (PDB:1SRY). The results with dotOmni (Figure 1a) show a strong agreement in predicting the approximate position of this domain, among the different parameter sets for the two algorithms. SEG assigns the entire non-globular domain and COILS2 identifies the

coiled-coil part that has heptad repeats. HISEG results are complementary to SEG and correspond to the globular domain. PSEG supports the presence of a heptad repeat by reporting a segment of period 7 in the same region, but not in any other period. An additional segment at residues 280-310 gives a positive signal with some of the parameter sets, particularly with COILS2 at its shorter window length settings of 14 and 21 (Figure 1a). However no PSEG segment is reported. The three-dimensional structure (Figure 1b) confirms that this segment is not a coiled-coil: it is actually a relatively amphiphilic surface alpha-helix of the globular domain. Figure 1c illustrates the greater structural mobility of the non-globular domain, suggested by the experimentally determined crystallographic temperature factors. This N-terminal domain, and also neighboring loops in the globular domain, makes a substantial conformational shift on binding tRNA (Biou *et al.* 1994).

In contrast with 1SRY, in xylose isomerase from *Streptomyces rubiginosus* (Figure 2), the C-terminal domain is known from the crystal structure (PDB:1XIS) to be a non-globular extension that wraps round another subunit of the tetramer, but this region does not contain any coiled-coil conformation. SEG identifies this structure as having relatively low compositional complexity, shown as segments together with an alignment of database matches (Figure 2a) and as highlighted region of the 1XIS structure (Figure 2b), whereas COILS2 gives negative results. This example illustrates the ability of SEG to identify a non-globular region on the general basis of sequence complexity data. In this case, there are no regular repeats or sequence patterns that can be modeled on the basis of a known structural class such as coiled-coil. Several other examples of relatively long, low complexity regions, that are identified by SEG in protein sequences of known crystal structure, correspond to parts of less well-determined electron density , or in many cases are “missing” from the crystallographic data, suggesting structural flexibility (Wootton 1994a; Wootton unpublished).

Combining coiled-coil and compositional complexity analysis can also be used to detect if other types of low complexity regions cause false prediction of coiled-coils. Figure 3 shows the blxOmni results for the *C. elegans* protein C25A11.4A. COILS2 gives a strong coiled-coil signal with most parameters, and SEG indicates low complexity. However, PSEG produces repetitive low complexity segments in a wide variety of periods, which suggests that the coiled-coil prediction was fooled by a strongly biased sequence composition. The residues reported by PSEG, visible in the Blixem window in figure 3, indicate that the region is very rich in glutamate and argninine. This “oversensitivity” to regions with charged residues was also noted by the authors of COILS2. This analysis further illustrates that COILS2 alone, in the absence of complexity and periodicity analysis, would probably give a misleading concept of the nature of this sequence.

Transmembrane analysis

A special case of biased sequence composition, are regions of the polypeptide that span a membrane. Because of the lipid environment, the protein is constrained to hydrophobic residues, particularly for alpha-helices that are exposed on all sides to the lipids. Transmembrane alpha helices that have hydrophilic interactions with other helices are generally less hydrophobic. Sophisticated transmembrane prediction programs improve accuracy by exploiting the difference in charged residues between loops on the cytoplasmic and non-cytoplasmic sides of the membrane (Claros and von Heijne 1994; Persson and Argos 1997; von Heijne and Gavel 1988). The most striking difference is the preference for the positively charged lysine and arginine on the cytoplasmic side. Incorporating such signals also allows the topology, i.e. the orientation relative to the cytoplasm, to be predicted.

Transmembrane segment prediction is not only important from a structural point of view, but also gives a strong indication of a protein's localization. Transmembrane prediction programs are prone to predict signal peptides as integral membrane segments.

Figure 4 shows the output of four transmembrane prediction programs of this type, TMHMM (Sonnhammer *et al.* 1998), HMMTOP (Tusnady and Simon 1998), MEMSAT (Jones *et al.* 1994), and PHDHTM (Rost *et al.* 1996), together with hydrophobicity curves for a given window length according to two scales (Black and Mould 1991; Kyte and Doolittle 1982). Also shown are signal peptide features from signalP (Nielsen *et al.* 1997), including the segment between the N-terminus and the most likely cleavage site if one is found within the first 50 residues. Positively charged residues (arginine and lysine) are marked as boxes to show whether they cluster on the cytoplasmic side. The hydrophobicity scales are highly correlated but can differ in some cases, e.g. the transmembrane segment around residue 300 in the example in figure 4 is much better supported by the Kyte-Doolittle curve than by the Black-Mould curve. The Dotter view features a dot-plot display of the query sequence vs. a randomly generated sequence of hydrophobic residues according to a distribution typical for transmembrane segments. This way, the Dotter Greyramp tool can be used to see the relative strength of transmembrane propensity for different regions.

In the example, rat glycine receptor beta chain precursor (SWISS-PROT: P20781), all four programs predict different topologies. However, looking vertically at individual TM segments, five of them are supported by three of the four methods, although by different sets of methods. The orientation N-in is also supported by three methods (assuming that the N-terminal segment predicted by HMMTOP is a cleaved signal peptide). TMHMM and MEMSAT predict fewer segments than the consensus, while HMMTOP is the only method that predicts the signal peptide as a transmembrane segment. The SWISS-PROT annotation is consistent with the TMHMM prediction, which lacks the segment around residue 90 that was predicted by the three other programs. Given the presence of a signal peptide, the SWISS-PROT/TMHMM topology appears correct. The segment around residue 90, predicted by the three other programs, is not strongly supported by the hydrophobicity curves or the dot-plot, and therefore probably represents a buried helix in a large extracellular domain. Because the N-terminal part of this globular domain contains clusters of positively charged residues, prediction algorithms can easily be fooled to force this part over to the cytoplasmic side for a better score. This example illustrates that taking the consensus prediction does not necessarily produce the correct prediction, but assisted with underlying propensities and dot-plots, the predictions can be validated and a correct result can be achieved.

Using the SFINX WWW server with Blixem and Dotter as helper applications

The analyses described here can be achieved without installing the assortment of back-end analysis programs and the SFINX package locally. It is sufficient to install the Dotter or Blixem viewers as helper applications to a web browser and run all prediction programs on the CGR web server at http://www.cgr.ki.se/SFINX . The web page allows the sequence complexity, structure, and transmembrane analyses to be turned on or off individually. The Blixem view can include BLAST results from a "netblast" search of the NR database at the NCBI. See instructions in the web page on how to set up Blixem and Dotter as helper applications.

METHODS

Programs and availability

The facilities described here are available in Blixem version 3.0 and Dotter version 3.0. Both these programs are written in C and use the ACEDB graphics library (Durbin and Thierry-Mieg 1999). Binaries are provided for X-windows on Unix workstations and Windows 95/98/NT. To parse output from BLAST to view in Blixem, a filtering program MSPcrunch is necessary. Dotter, is available at ftp://ftp.cgr.ki.se/pub/prog/dotter. Blixem and MSPcrunch are available at the same ftp servers, but in the directory MSPcrunch+Blixem instead of dotter. Documentation on how to run Dotter and Blixem with SFS data can be found at http://www.cgr.ki.se/cgr/groups/sonnhammer/Dotter.html and Blixem.html. Both allow the user to control the display of the features with a "Feature Series Selection Tool" (Figure 1a) in which each series can be individually turned on or off. The capability of selectively showing the series is crucial since by default a large number of series are generated, of which normally only a few are relevant at one time.

The connection to external programs was done by csh and gawk scripts (see Table 1) which are available at ftp://ftp.cgr.ki.se/pub/prog/SFINX.

The SEG, PSEG, HISEG, and NSEG programs are freely available at ftp://ncbi.nlm.nih.gov/pub/seg. COILS2 was downloaded from ftp://alf.biochem.mpg.de/Coils.

See http://www.cbs.dtu.dk/services/TMHMM/ for acquiring TMHMM, http://www.enzim.hu/hmmtop/ for HMMTOP, http://insulin.brunel.ac.uk/~jones/memsat.html for MEMSAT, and ftp://cubic.bioc.columbia.edu/pub/rost/ for PHD.

A generic sequence feature series data format

The 'SFS' data format is a 'meta-language' between non-graphical computation programs and graphical viewers that allows generic data exchange for visualization purposes. To make it as universal as possible, the format mainly supports only the core data. Information such as screen placement of the objects, fonts, order of the series, etc., were explicitly avoided, since such features are better controlled interactively in the graphical viewer. The SFS specification follows below. By "data point" we mean the smallest unit of data, either a segment or an XY-value pair in a curve.

1. Each data point is associated with a named *series*; one series can contain any number of data points of any number of data types.

2. Each data point is stored on one line (<10000 characters).

3. The fields in a line are separated by white-space characters, which are not allowed within fields.

4. SFS data should be preceded by a header line with the words "# SFS format 1.0" for backwards compatibility in the future.

5. SFS data should be preceded by a data type specifier; currently one of:
   # SFS type=SEG
   # SFS type=XY <data…>
   # SFS type=HSP
   # SFS type=GSP
   # SFS type=GFF
   # SFS type=SEQ <data…>

   The SEG type specifies that segment data follow; XY that curve (XY plot) data follow; HSP ("High Scoring Pair", as in the BLAST programs) indicates that ungapped pairwise matches follow; and GSP ("Gapped High Scoring Pair") that gapped matches follow. GFF is included for compatibility with the existing GFF format. SEQ data is the amino acid or nucleotide sequence from which the SFS data was generated: visualization tools often require the original sequence. For segment data (SEG, HSP, GSP, and GFF), properties such as color and annotation is given per segment, while for data of XY type these are given once for an entire curve. All XY coordinates are considered to belong to one curve until the next "# SFS type=" line. However, one XY series can contain any number of curves.

6. For data of type SEG (segment data), the format of each segment is:
   <score> <seqname> <seriesname> <start> <end> <look> [annotation]
   These fields are specified as:

<score>	[int] The score of the segment[1]
<seqname>	[string] The sequence that the feature belongs to[2]
<seriesname>	[string] Name of series that this data belongs to
<look>	[string, comma separated list in one word] The appearance, e.g. color[3]
<start>	[int] Start coordinate of segment
<end>	[int] End coordinate of segment.
[annotation]	[strings] Optional description of the segment

7. For data of type XY (curves), the format of the type specifier is:
   # SFS type=XY <seqname> <seriesname> <look> [annotation]
   where the fields are the same as specified for the segment data under 6.
   All lines until the next "# SFS type=" line must contain XY data, of which the format is:
   <x> <y>
   Specified as:
   <x> [int] Residue number in sequence
   <y> [int] Y-value at residue x[1]

8. For data of type HSP:
   <score> <qname> <qframe> <qstart> <qend> <sname> <sframe> <sstart> <ssend> <sequence>
   These fields are specified as:

<score>	[int, 0-100] The score of the segment
<qname>	[string] Name of the query sequence
<qframe>	[string] Frame of the query segment, "+1", "+2", "+3", "-1", "-2", "-3"
<qstart>	[int] Start coordinate of query segment
<qend>	[int] End coordinate of query segment
<sname>	[string] Name of subject sequence

<sframe> [string] Frame of subject segment
<sstart> [int] Start coordinate of subject segment
<ssend> [int] Start coordinate of subject segment
<sequence> [string] Sequence of matching subject segment

The annotation of each sequence may be given on the next line, preceded by "# DESC ".

9. For data of type GSP:
   <score> <qname> <qframe> <qstart> <qend> <sname> <sframe> <sstart> <ssend> <sequence>
   where the fields are the same as specified for the segment data under 8.
   All lines until the next "# SFS type=" line contain the gapped pairwise alignment, in the form of pairwise starting points and lengths of each ungapped segment (block). It is assumed that regions between ungapped blocks contain an insertion in one sequence only, while the other sequence has a zero distance between two adjacent blocks. The format to specify each ungapped block is:
   <qstart> <sstart> <len>
   Specified as:
   <qstart> [int] Starting point in query sequence.
   <sstart> [int] Starting point in matching database sequence (subject).
   <len> [int] Length of the ungapped block (number of residues).

10. For data of type GFF:
    <seqname> <seriesname> <look> <start> <end> <score> <strand> <transframe> [annotation]
    where the fields are the same as specified for the segment data under 6, except:
    <strand> [char] For DNA, the strand '+', '-', or '.'.
    <transframe> [int] For coding DNA, the frame of the codons. '0', '1', '2', or '.'.
    See (http://www.sanger.ac.uk/Software/GFF) for details on the GFF format.

11. For data of type SEQ, the format of the type specifier is:
    # SFS type=SEQ <sequence> <seqname> [annotation]
    where the fields are the same as specified for XY data under 7, except:
    <sequence> @[int] Ordinal number of provided sequence preceded by '@': "@1", "@2", etc.. This is necessary when multiple sequences are included, for instance for Dotter.
    All lines until the next "# SFS type=" line contain the entire query sequence. No formatting characters should be used in the sequence.

Footnotes (including implementation-specific details in Blixem and Dotter):

1. For simplicity, the score is required to fall between 0 and 100; the raw score must thus be rescaled. In many cases, it is wise to rescale the score so that the 'twilight zone' scores fall in the 0-100 range, while all clearly significant scores are converted to 100. This is advantageous for visualization purposes, as it focuses the analyst's attention to features that require critical evalutation. The actual score of clearly significant features is normally not important.
2. Blixem and Dotter can for simplicity use special shorthand codes for the field <seqname>. "@1" means the horizontal sequence, and "@2" the vertical sequence.

3. The <look> field contains information of the appearance of a particular feature, e.g. its color, shape, line thickness, etc. Multiple attributes are allowed to be specified as comma separated lists, in which the attributes are concatenated to one word with a single comma character as separator (no space before or after the comma). The exact wording and meaning of the look attributes need to be specified in the definition as "magic tags". In SFS 1.0, SEG data are restricted to colors, and XY data to color and a drawing mode ("interpolated" or "partial"). By default, XY curves are linearly interpolated in regions where no data was given, but if "partial" is used in the look field, the curve is only drawn in the specified regions. Interpolation greatly simplifies the specification of straight lines which are commonly used for indicating thresholds etc. The colors in Dotter and Blixem are limited to the color names used by ACEDB, which allows 32 common colors (see http://www.cgr.ki.se/cgr/groups/sonnhammer/Dotter.html). Aside from parital/interpolated, no shape attributes are specified in SFS 1.0. Alternative shapes might be useful, particularly for DNA sequence features such as introns and splice sites, but on the other hand these can easily be accommodated as XY curves.
4. For backwards compatibility, Blixem also still supports the old SEQBL format (<score> <qframe> <qstart> <qend> <sstart> <ssend> <sname> <sequence>), which is a simpler version of the generic HSP data type specification.
5. GSP data is currently not supported in MSPcrunch and Blixem. However, MSPcrunch can turn gapped HSPs from gap-BLAST into pseudo-ungapped HSPs, in which deletions in the subject sequence are shown as gaps while insertions are collapsed. These can be displayed in Blixem with almost no loss of information.

An example containing some segments and a curve to mark up a 200 residue long sequence is shown below. A threshold line is specified in the last three lines.

```
# SFS format 1.0
# SFS type=SEG
100 @1 TM 1 38 yellow TM prediction: brown=TM, yellow=cytoplasmic
100 @1 TM 39 61 brown
100 @1 TM 62 73 white
100 @1 TM 74 99 brown
100 @1 TM 100 112 yellow
100 @1 TM 113 133 brown
100 @1 TM 134 152 white
# SFS type=XY @1 myhydrophob green My hydrophobicity
1 10
36 10
41 90
59 90
64 10
71 10
76 90
97 90
102 10
110 10
115 90
131 90
136 10
152 10
```

```
# SFS type=XY @1 myhydrophob black
1 50
200 50
```

DISCUSSION

The main conclusion from this work is that the picture of sequence features becomes clearer as more types of analyses and more parameter combinations are explored. Many analysis methods have been developed using proteins of 'typical' amino acid composition and may produce highly misleading results when applied to protein sequences of 'atypical' composition, i.e. strongly biased towards a few amino acids. Therefore it is valuable to also look at sequence composition directly, to be able to judge whether a feature prediction may have been influenced by biases in sequence composition. Many programs for predicting coiled coil and transmembrane are prone to produce mispredictions on sequences of biased composition. We provide a set of general rules for assigning structural class based on compositional features in table 2; it is however important to keep in mind that all of the features are only indicative and not conclusive. They need to be judged in the context of local sequence composition biases and repeats in order to avoid false predictions. This context is provided by the graphical SFINX package described here.

The viewers in the SFINX package employ the SFS data exchange format for importing predictions and data from a variety of sequence analysis programs. The SFS data format is meant to be a generic vehicle for exchanging sequence features including curves, functioning as a meta-language between computing programs and graphical viewers. We believe that such a system will accelerate the development of future computation programs, because providing such programs with an interface for the simple SFS format is clearly easier than developing a entirely new viewer. It may also stimulate development of more sophisticated and interactive results viewers. We hope that an SFS viewer will soon be available in Java; in the mean time, Dotter and Blixem can be used as WWW helper applications under UNIX X-windows and Windows 95/98/NT.

The SFS format currently fulfills the requirements of the most fundamental generic tasks. There are a number of more specialized tasks that would profit from a special data type, which we have not supported here. One example is symbols for gene finding, where splice sites and introns usually have a different layout than the common boxes. This could in principle be indicated with <look> attributes specifying shapes, e.g. intron, splice5, or arrow. However, to keep the SFS format as generic and as simple as possible, and as most shapes can be well represented by XY curves, we have refrained from defining these looks here. In most cases it is however sufficient to mark up features with a particular meaning using special color codes.

One consequence of the SFINX package's design is that the scripts are pre-configured with certain parameter choices. These are thus not interactively settable, but we believe that our selection of parameters in the release and on the web server will serve casual users well. More advanced users will need to download the scripts and analysis programs, and can then easily modify the parameters to their own choice.

ACKNOWLEDGEMENTS

We would like to thank two anonymous referees for useful suggestions.

REFERENCES

Altschul SF, Boguski MS, Gish W, Wootton JC. 1994. Issues in searching molecular sequence databases. Nature Genetics 6:119-129.

Bailey LC, Jr., Fischer S, Schug J, Crabtree J, Gibson M, Overton GC. 1998. GAIA: framework annotation of genomic sequence. Genome Res 8:234-250.

Bairoch A, Apweiler R. 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 28:45-48.

Biou V, Yaremchuk A, Tukalo M, Cusack S. 1994. The 2.9 A crystal structure of *T. thermophilus* seryl-tRNA synthetase complexed with tRNA(Ser). Science 263:1404-1410.

Bisson G, Garreau A. 1995. APIC: a generic interface for sequencing projects. ISMB 3:57-65.

Black SD, Mould DR. 1991. Development of hydrophobicity parameters to analyze proteins which bear post- or cotranslational modifications. Anal Biochem 193:72-82.

Claros MG, von Heijne G. 1994. TopPred II: an improved software for membrane protein structure prediction. Comput. Appl. Biosci. 10:685-686.

Dunker AK, Garner E, Guilliot S, Romero P, Albrecht K, Hart J, Obradovic Z, Kissinger C, Villafranca JE. 1998. Protein disorder and the evolution of molecular recognition: theory, predictions and observations. Pac Symp Biocomput:473-484.

Durbin R, Thierry-Mieg J. 1999. The ACEDB genomic database. World Wide Web URL: ftp://ftp.sanger.ac.uk/pub/acedb.

Fenyo D. 1999. The biopolymer markup language. Bioinformatics 15:339-340.

Jones DT, Taylor WR, Thornton JM. 1994. A model recognition approach to the prediction of all-helical membrane protein structure and topology. Biochemistry 33:3038-3049.

Karlin S, Brendel V. 1992. Chance and statistical significance in protein and DNA sequence analysis. Science 257:39-49.

Kyte J, Doolittle RF. 1982. A simple method for displaying the hydropathic character of a protein. J Mol Biol 157:105-132.

Lupas A. 1996. Prediction and analysis of coiled-coil structures. Methods In Enzymology 266:513-525.

Lupas A, van Dyke M, Stock J. 1991. Predicting coiled coils from protein sequences. Science 252:1162-1164.

Nielsen H, Engelbrecht J, Brunak S, von Heijne G. 1997. A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Int J Neural Syst 8:581-599.

Persson B, Argos P. 1997. Prediction of membrane protein topology utilizing multiple sequence alignments. J Protein Chem 16:453-457.

Rost B, Fariselli P, Casadio R. 1996. Topology prediction for helical transmembrane proteins at 86% accuracy. Protein Sci 5:1704-1718.

Rubin G. 1996. Around the Genomes: The *Drosophila* Genome Project. Genome Res. 6:71-79.

Salomon P, Konopka AK. 1992. A Maximum Entropy Principle for Distribution of Local Complexity in Naturally Occurring Nucleotide Sequences. Computers chem. 16:117-124.

Searls DB. 1995. bioTk:componentry for genome informatics graphical user interfaces. Gene 163:GC1-16.
Sonnhammer ELL, Durbin R. 1994. A workbench for large-scale sequence homology analysis. Comput. Appl. Biosci. 10:301-307.
Sonnhammer ELL, Durbin R. 1995. A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. Gene 167:GC1-10.
Sonnhammer ELL, von Heijne G, Krogh A. 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. ISMB 6:175-182.
Spitzner JH. 1999. Bioinformatic Sequence Markup Language (BSML). World Wide Web URL: http://www.visualgenomics.com/bsml/.
Sussman JL, Lin D, Jiang J, Manning NO, Prilusky J, Ritter O, Abola EE. 1998. Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules. Acta Crystallogr D Biol Crystallogr 54:1078-1084.
Tusnady GE, Simon I. 1998. Principles governing amino acid composition of integral membrane proteins: application to topology prediction. J Mol Biol 283:489-506.
von Heijne G, Gavel Y. 1988. Topogenic signals in integral membrane proteins. Eur J Biochem 174:671-678.
Wootton JC. 1994a. Non-globular domains in protein sequences: automated segmentation using complexity measures. Comput. Chem. 18:269-285.
Wootton JC. 1994b. Sequences with 'unusual' amino acid compositions. Curr opin struct biol 4:413-421.
Wootton JC, Federhen S. 1993. Statistics of local complexity in amino acid sequences and sequence databases. Comput. Chem. 17:149-163.
Wootton JC, Federhen S. 1996. Analysis of compositionally biased regions in sequence databases. Methods Enzymol 266:554-571.
Wright PE, Dyson HJ. 1999. Intrinsically unstructured proteins: re-assessing the protein structure- function paradigm. J Mol Biol 293:321-331.
Zhang J, Ostell J, Rudd K. 1994.ChromoScope: a graphic interactive browser for *E. coli* data expressed in the NCBI data model. Proc. 27th Annual Hawaii International Conference on System Sciences. 58-67 p.

TABLES

Table 1. List of the scripts in the package presented here that are coupled to Dotter and Blixem analysis. Note that all Blixem displays can be combined with output from BLAST. All Dotter scripts except blxTM produce a self-dot-plot; blxTM makes a dot-plot of the query sequence and a randomly generated hydrophobic sequence.

	Front-end scripts	**Runs programs**	**Helper scripts**
Sequence complexity analysis by SEG with multiple parameter sets	blxseg dotseg	seg pseg hiseg	SFSseg seg2SFS pseg2SFS SFSentropy
Secondary structure and accessibility prediction by PHD and coiled coil prediction by COILS2 with multiple parameter sets	blxStruct dotStruct	phd sec phd acc coils2	SFSstruct phdsec2SFS phdacc2SFS coils2SFS coils2script
Transmembrane prediction by TMHMM, HMMTOP, MEMSAT, and PHDHTM; hydrophobicity plots	blxTM dotTM	tmhmm hmmtop memsat phd htm signalp	SFSTM TMHMM2SFS HMMTOP2SFS memsat2SFS phdhtm2SFS signalp2SFS signalp2seq hydroph
Integrated complexity, coiled-coil, and transmembrane analyses	BlxOmni DotOmni	All of the above	All of the above

Table 2. Guidelines for interpreting sequence composition derived features to assign the structural class of a protein. The interpretation is significantly enhanced if these features are analyzed graphically in the context of dot-plots and matching sequences with the SFINX package.

Structural class	Positive indications	Negative indications
Non-globular, type coiled coil	- Coiled-coil support with many parameter sets. - SEG low complexity. - PSEG low complexity of period 7.	- Not supported by many coiled-coil parameter sets. - HISEG high complexity. - PSEG low complexity of other periods than 7.
Non-globular, other types	- SEG low complexity. - PSEG low complexity of various periodicities.	- HISEG high complexity.
Transmembrane	- TM prediction supported by many methods. - Supported by hydrophobicity propensities.	- SEG or PSEG low complexity. - Overlap with signal peptide prediction.

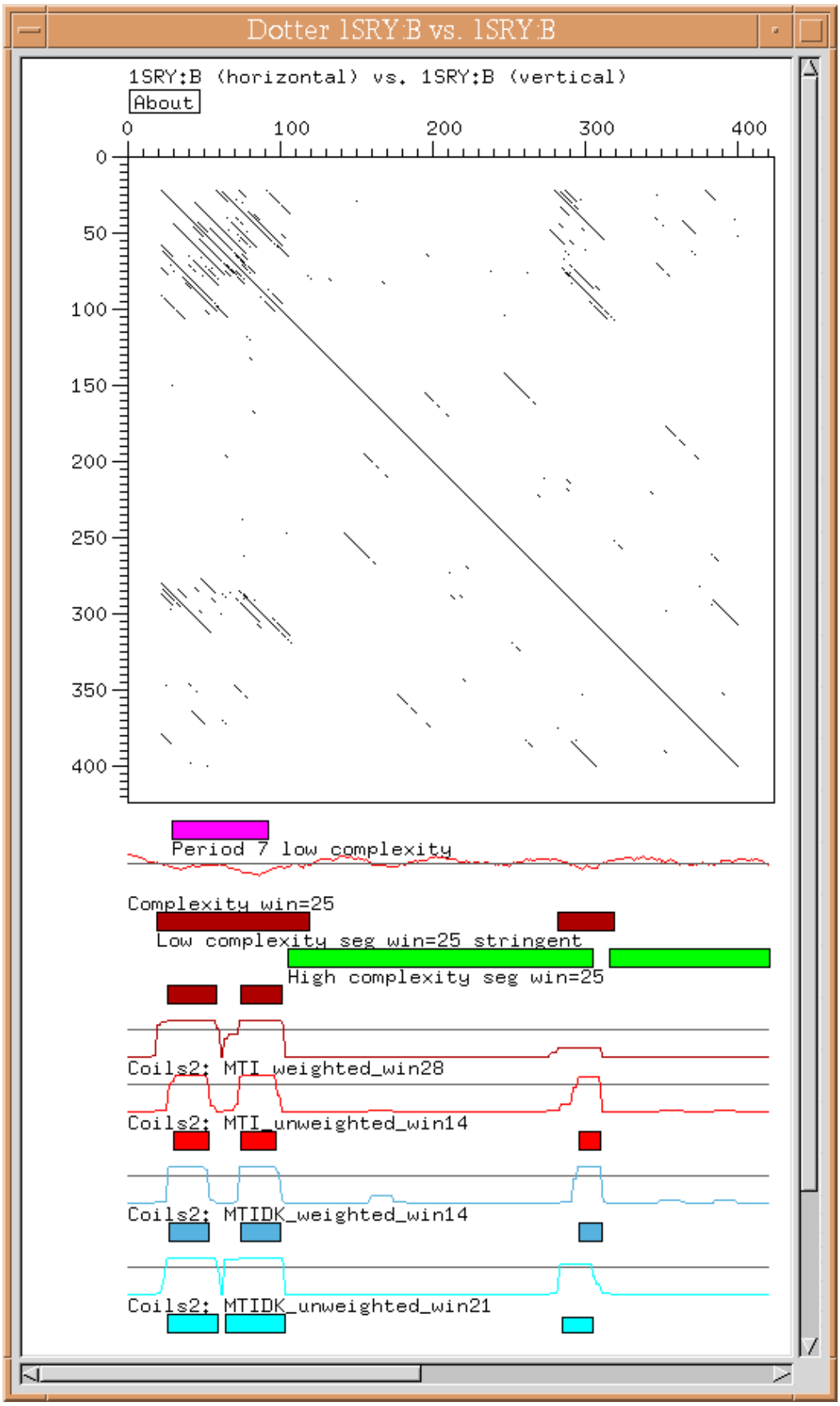
FIGURES

Figure 1. Combined coiled-coil and sequence complexity analysis applied to seryl tRNA synthetase (PDB:1SRY). A. The graphical output in Dotter produced by the dotOmni script. Relevant feature series from PSEG, SEG and COILS2 were selected. The dot-plot was calculated with a window size of 42. B. Actual structure of one monomer in 1SRY. The segments found by SEG low complexity analysis and COILS2 are marked dark in the structure. The extended N-terminal region, which is found with all parameter settings, is a typical coiled-coil. However, the short segment detected around residue 300 is not a coiled-coil, but merely an amphipathic surface helix. Indications that this was a false positive prediction include the facts that no PSEG low complexity segment of period 7 was found in this region, and that only some COILS2 parameter settings predict it. Such short spurious predictions are rather common, hence only looking at one of the coiled-coil predictions might give a misleading result. C. The 1SRY structure colored according to the crystallographic temperature factors. High temperature (flexible) residues are dark while rigid residues are light. The flexible region corresponds to the N-terminal segment predicted as coiled-coil.

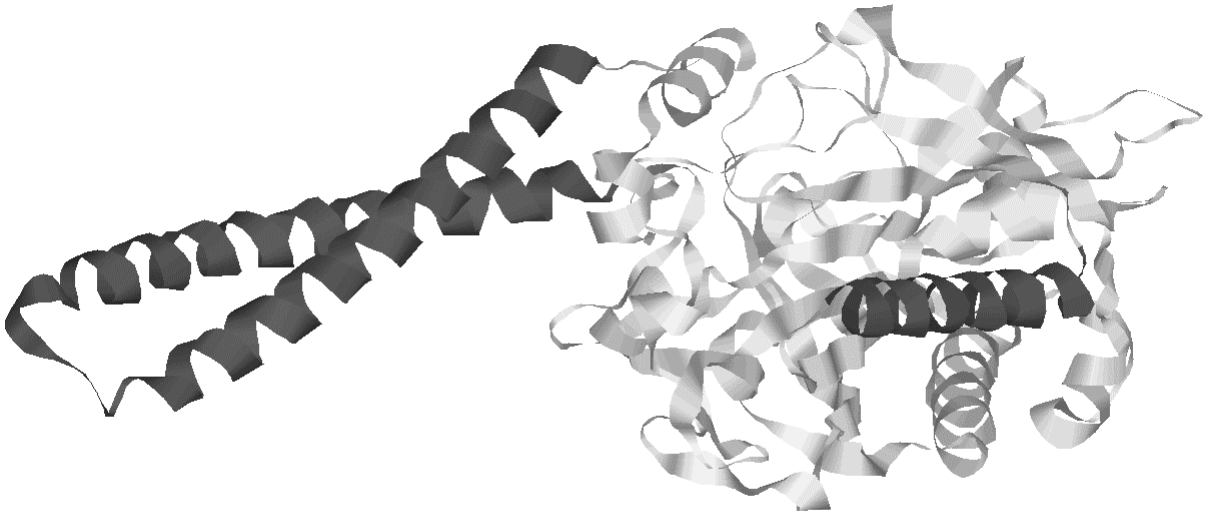
Figure 2. Analysis if non-globular segments that are not coiled-coils, applied to xylose isomerase (PDB:1XIS). A. Blixem display of results produced by the blxseg script (selected feature series shown) together with database matches reported by BLAST. The C-terminal region is found to have low sequence complexity, suggesting that it has an irregular, non-globular structure. This region is only present in some homologs. B. Actual structure of one monomer in 1XIS. The low complexity segment found by SEG is marked dark and corresponds to an extended, non-globular tail with a partly irregular structure. Because this extended segment is not of the coiled-coiled type, COILS2 does not detect it.

Figure 3. Coiled-coil or not? SFINX analysis of *C. elegans* protein C25A11.4A. The blxOmni output shows that the region predicted by COILS2 to contain a coiled coil, also features very low sequence complexity in various periodicities, as reported by PSEG. The coiled-coil prediction is unlikely to be correct because the region is much more biased towards charged residues than a typical coiled-coil, and there is no preference for low sequence complexity in period 7. A more likely scenario is that this is a charged cluster with a flexible or irregular folding pattern.

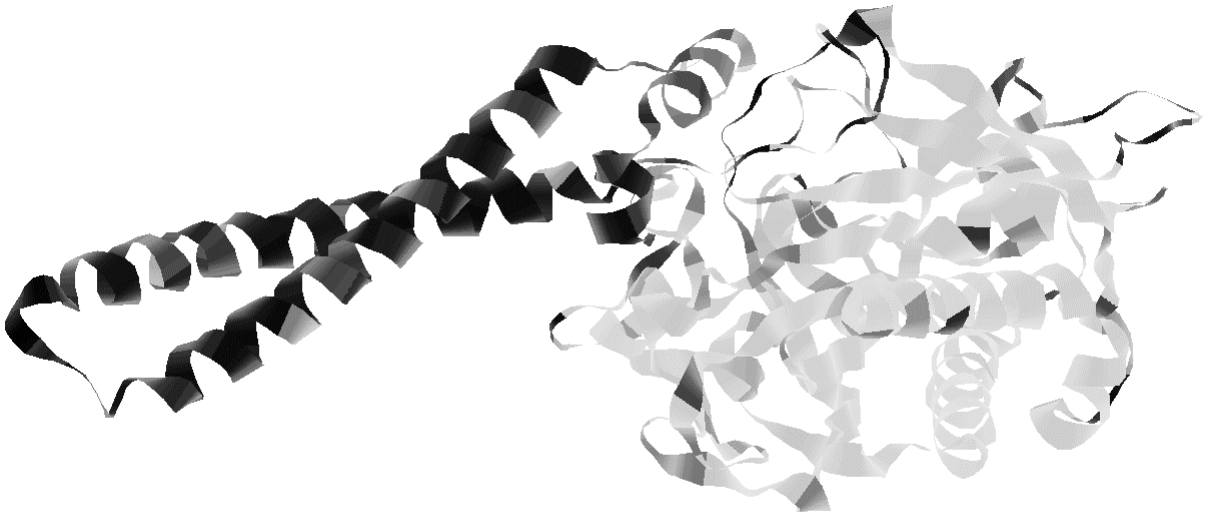
Figure 4. Combined transmembrane topology analysis, applied to a glycine receptor (SWISS-PROT:GRB_RAT). The dotTM output shows a dot-plot of the query sequence versus a randomly generated sequence of hydrophobic residues, along with the results from signalP and four TM prediction methods (written below the prediction), followed by positively charged residues and hydrophobicity curves from two different scales. The topology predictions are marked according to: dark=in the membrane; shaded=cytoplasmic loop; white=non-cytoplasmic loop. All four predictions are different - which one is correct? The dotplot and hydrophobicity curves indicate that the TMHMM prediction is most likely the correct one.


Dotter 1SRY:B vs. 1SRY:B
1SRY:B (horizontal) vs. 1SRY:B (vertical)
About
0
100
200
300
400
0
50
100
150
200
250
300
350
400
Period 7 low complexity
Complexity win=25
Low complexity seg win=25 stringent
High complexity seg win=25
Coils2: MTI_weighted_win28
Coils2: MTI_unweighted_win14
Coils2: MTIDK_weighted_win14
Coils2: MTIDK_unweighted_win21

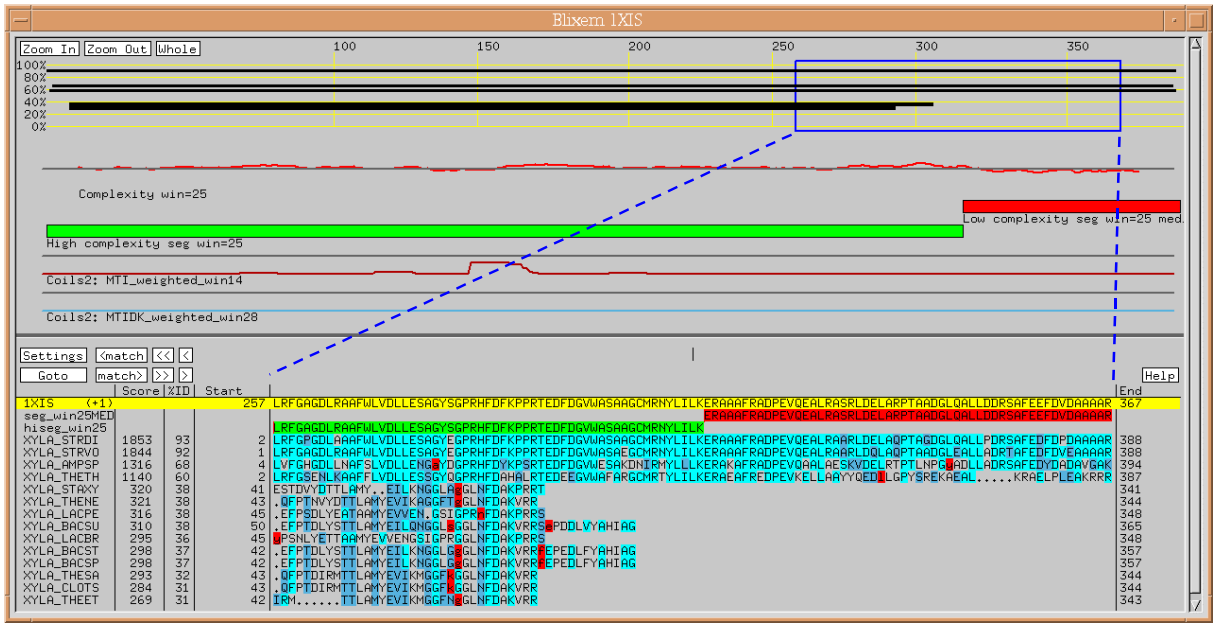
1A



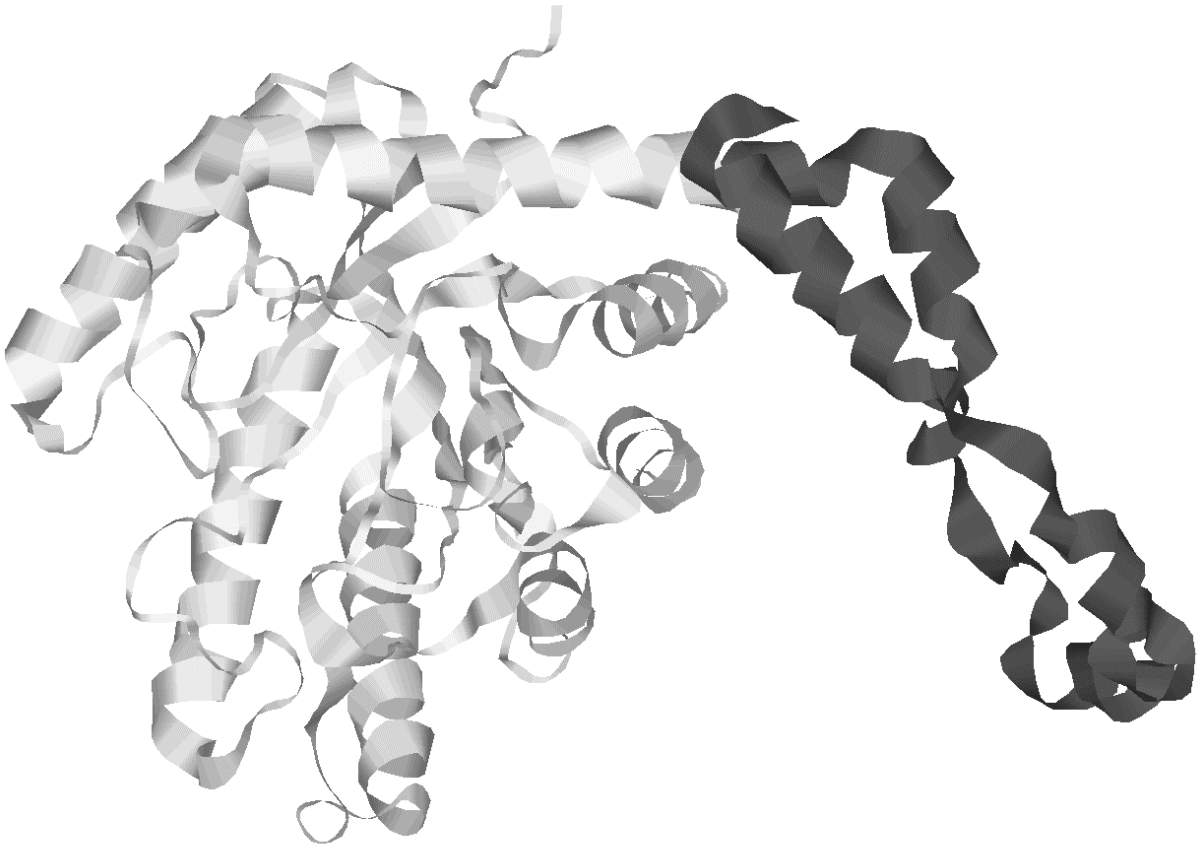
1B



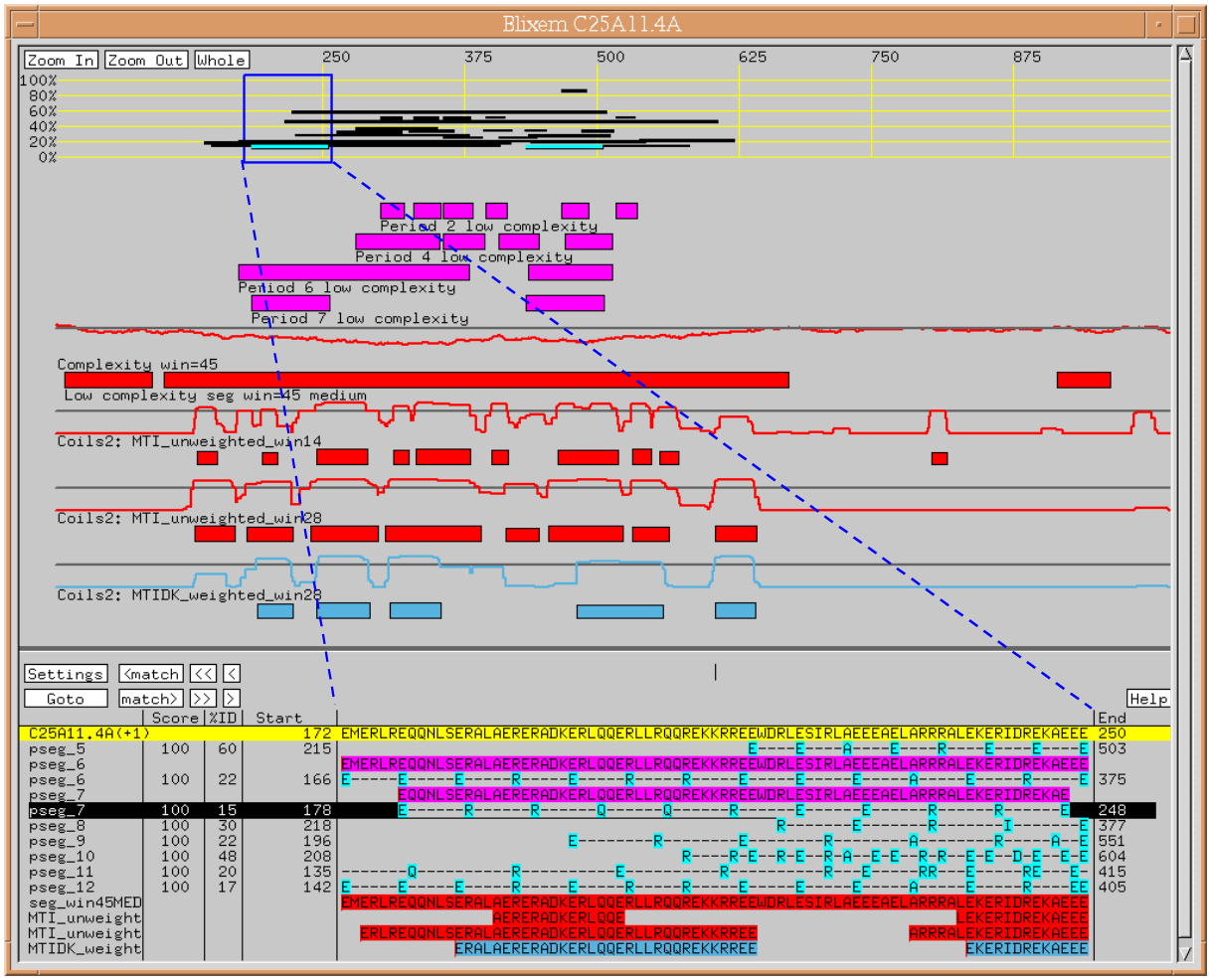
1C


Blixem 1XIS
Zoom In
Zoom Out
Whole
Complexity win=25
High complexity seg win=25
Low complexity seg win=25 med
Coils2: MTI_weighted_win14
Coils2: MTIDK_weighted_win28
Settings
Goto
Help

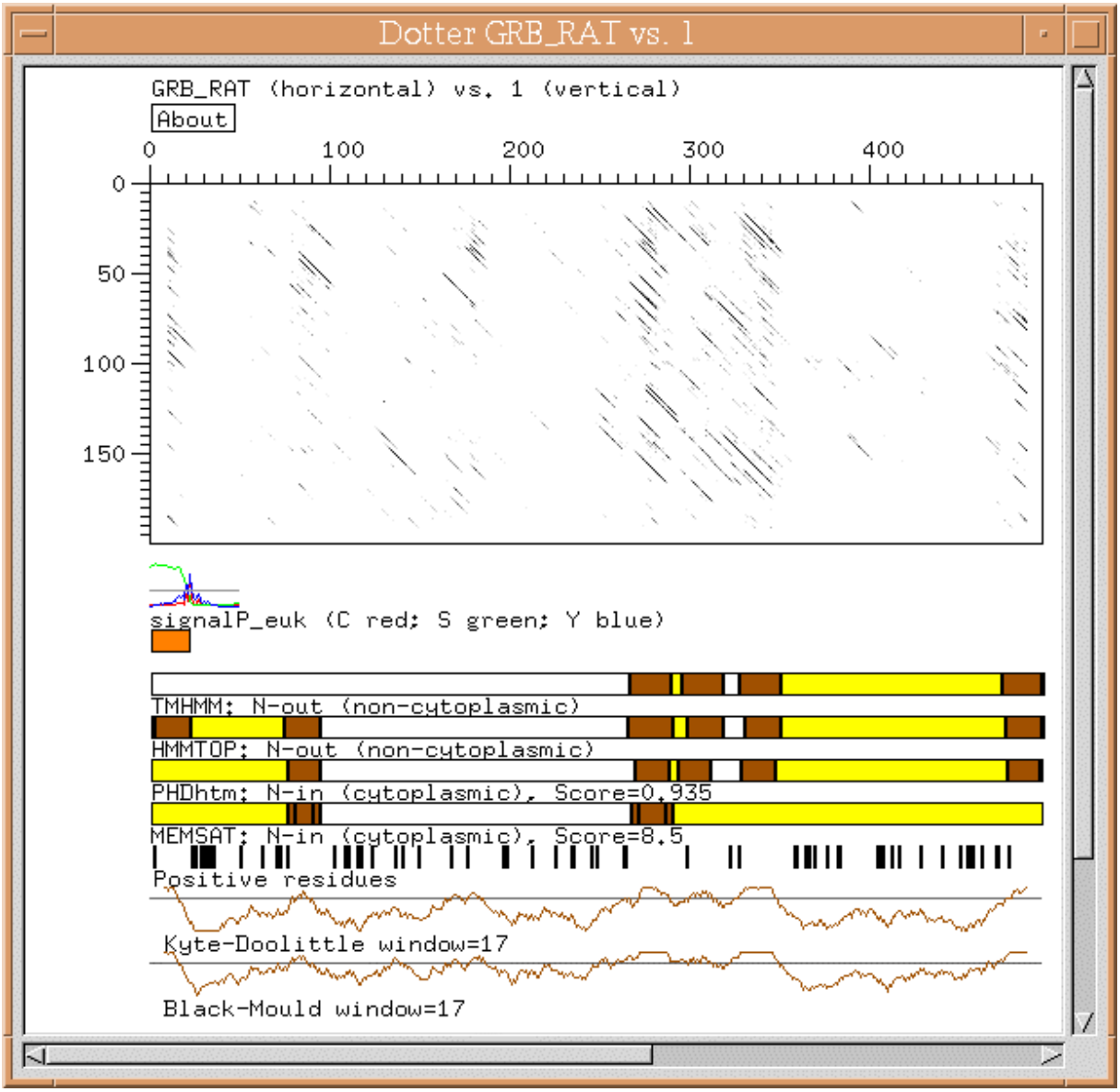
2A



2B


Blixem C25A11.4A
Zoom In
Zoom Out
Whole
250
375
500
625
750
875
100%
80%
60%
40%
20%
0%
Period 2 low complexity
Period 4 low complexity
Period 6 low complexity
Period 7 low complexity
Complexity win=45
Low complexity seg win=45 medium
Coils2: MTI_unweighted_win14
Coils2: MTI_unweighted_win28
Coils2: MTIDK_weighted_win28
Settings
<match
<<
<
Goto
match>
>>
>
Help
Score
%ID
Start
End
C25A11.4A(+1) 172 EMERLREQQNLSERALAERERADKERLQQERLLRQQREKKRREEWDRLESIRLAEEEAELARRRALEKERIDREKAEEE 250
pseg_5 100 60 215 503
pseg_6
pseg_6 100 22 166 375
pseg_7
pseg_7 100 15 178 248
pseg_8 100 30 218 377
pseg_9 100 22 196 551
pseg_10 100 48 208 604
pseg_11 100 20 135 415
pseg_12 100 17 142 405
seg_win45MED
MTI_unweight
MTI_unweight
MTIDK_weight

3


Dotter GRB_RAT vs. 1
GRB_RAT (horizontal) vs. 1 (vertical)
About
0
100
200
300
400
0
50
100
150
signalP_euk (C red; S green; Y blue)
TMHMM: N-out (non-cytoplasmic)
HMMTOP: N-out (non-cytoplasmic)
PHDhtm: N-in (cytoplasmic), Score=0.935
MEMSAT: N-in (cytoplasmic), Score=8.5
Positive residues
Kyte-Doolittle window=17
Black-Mould window=17