

SPECIFIC MODELLING OF REGULATORY UNITS IN DNA SEQUENCES

K. FRECH, T. WERNER

GSF - National Research Center for Environment and Health, Institute of Mammalian Genetics, Ingolstädter Landstraße 1, D-85764 Neuherberg, Germany

Transcriptional control regions are usually composed of a complex arrangement of individual transcriptional elements like protein binding sites. This modular structure allows generation of enormous functional diversity of regulatory regions with a limited set of individual elements. We implemented simple formal representations of these general features of regulatory regions into an algorithm capable of developing complex models reflecting both the element composition and the functional organization of individual elements. Our method (ModelGenerator) requires a training set of at least 10 sequences containing the regulatory regions to be modelled and a very simple initial model which may consist of just two characteristic transcription factor binding sites. We show the capability of our algorithm to expand the initial model solely by comparative sequence analysis leading to complex, biologically meaningful models. A second program (ModelInspector) is capable to scan new sequence data for matches to models defined by ModelGenerator. We show two models for retroviral transcriptional control regions to be highly specific. A search against GenBank using one of the models is shown to be free of false negatives and to produce less than 2 false positives / million nucleotides. Thus, our algorithms appear to be useful tools for the analysis of extremely long genomic sequences which are now becoming available as results of various genome sequencing projects.

1 Introduction

The timely appearance of proteins at correct locations is crucial for development as well as functionality of higher organisms. Transcriptional regulation plays a major role in expression control. It has been shown for the role of engrailed-1 and engrailed-2 genes in mouse brain development that correct transcriptional regulation can be even more important than the protein itself [1]. Transcription is regulated by the interaction of proteins with their cognate DNA binding sites. These DNA elements are organized into regulatory units known as promoters, enhancers or silencers (including scaffold attachment regions, SARs and locus control regions, LCRs). Individual regulatory units are usually determined by the spatial organization of binding sites for common and cell- or tissue-specific transcription factors (TF-sites). These sites are separated by spacer-DNA which is usually not conserved precluding global alignment strategies for identification of regulatory units. At least promoters of functionally related gene families appear to have a detectable common organization of TF-sites as has been shown for *saccharomyces cerevisiae* [2,3].

Several methods for identification of individual TF-sites have already been developed and are available as programs [4,5,6,7]. However, transcriptional regulation is not an inherent property of individual binding sites but is determined by the context of binding sites. A simple statistical model of limited specificity based on the frequency profile of TF-sites in promoter and non-promoter sequences has been successfully implemented in a program to locate polymerase II promoters (PromoterScan, [8]). The program FunsiteP [9] uses the uneven distribution of TF-sites in promoter subregions for promoter recognition. A model free approach to assess the context by distance correlation of different TF-sites was shown to yield some basic organizational features of yeast promoters [2,3]. However, these basic patterns are not sufficiently specific for an unambiguous definition of promoters.

Other model-building approaches like GeneLang [10], GeneParser [11], or GRAIL [12] focus on gene prediction. Only GRAIL includes few basic features of the promoter regions in the last release.

Here, we present a new method for definition (ModelGenerator) and recognition (ModelInspector) of regulatory regions based on the concept of a strictly modular design of such units. ModelGenerator develops a model by combining several types of individual elements with their overall spatial organization within the regulatory unit. One of the hallmarks of the method is development of a complex model by sequence analysis which can reveal new unknown features not present in the initial model. ModelInspector is capable to scan unlimited sequences with such a model and showed an extraordinary high specificity for the models tested.

2 System and Algorithm

2.1 Basic requirements for model generation

ModelGenerator requires a set of at least 10 sequences (with moderate or low overall similarity) containing a single type of regulatory unit. This can be a set of promoter sequences of a gene family that shows common regulation or a set of homologous promoters from different species (e.g. within higher eukaryotes). In addition to the sequence set a very simple initial model (e.g. two characteristic TF-sites and their sequential order) is required.

2.2 Generation of a model

ModelGenerator defines two types of individual elements. *Determining elements* are part of the basic organizational framework of the regulatory unit (e.g. TATA box and/or initiator for TATA box containing promoters). They have to be present in

almost all of the sequences (few exceptions are allowed) and can be detected with a tolerable number of false positive matches, e.g. by one of the TF-site search programs. *Non-determining* elements are not *per se* clearly associated with the regulatory unit and are distinguished solely by association with other elements. Examples of such elements include poorly defined TF-sites or low energy hairpin structures.

The first step of the model development process is the location of all elements which are part of the initial model in all sequences. These elements are by definition determining elements. This allows to define a basic framework in the individual sequences which is composed of the sequential order of the initial elements and the regions in between. A region is the sequence from one determining element to another (or the sequence before the first or after the last determining element). This also allows the definition of a consistency criterion. An element is considered consistent with the basic framework if it occurs in all or almost all sequences in the same region. Further potential consensus elements are then identified by analysis of the sequence set by the program CoreSearch [13]. Elements found consistently in the majority of sequences are selected for the model as determining elements, whereas new elements not consistent with the basic framework are discarded.

All potential determining elements are located either by a consensus search [5,7] or by other search routines included in ModelGenerator (e.g. secondary structure elements, or direct repeats). Potential matches are rated according to their score assigned by the search program. For each sequence of the training set ModelGenerator then identifies all possible combinations of determining elements which reach the user-defined thresholds and occur in the same sequential order (D-sets). At this point, more than one potential D-set is still possible in a single sequence. The *basic set* is build of all sequences with a single D-set (at least 6 sequences are required). Distance histograms for adjacent elements are then generated from the basic set. A single D-set within sequences with more than one potential D-set can then be selected by the maximum element score and the maximum distance score based on the distance histograms. These sequences are added to the basic set. The determining elements and the regions define the basic framework of the regulatory unit model.

Analysis of individual sequences for non-determining elements is restricted to the regions as defined above. Elements within one region are selected if present in at least a significant subset of the sequences. Thus, all non-determining elements compatible with the basic framework are defined.

Finally, a complete set of distance histograms is calculated for all elements. The complex regulatory unit model consists of descriptions of all individual elements, their mean scores, their sequential order and the distance distributions observed in the training set.

2.3 *Detection of matches to the generated model*

Another program designated ModelInspector utilizes these models to scan any sequence for unknown regulatory units matching the model. First all matches for individual determining elements which score above the thresholds defined by the model are located in the new sequence. These individual elements are combined to match the organization of the D-set characteristic of the model. If the sum of scores of determining elements exceeds a given threshold (default: 50% of the average sum of the training set), non-determining elements are identified and the sum of scores of all elements is calculated. This total element score and the score for element distances are then used to evaluate the fit of complex elements to the model. Score thresholds are expressed in percent of the mean scores of the model (defaults are: 50% of average element score, and 30% of average distance score). An extensive description of the algorithm and the programs ModelGenerator and ModelInspector will be presented elsewhere (Frech *et al.*, in preparation).

3 Results

We selected retroviral transcriptional control regions in order to test the abilities of ModelGenerator and ModelInspector for several reasons. Retroviruses are expressed under the control of viral control regions designated long terminal repeats (LTRs), which contain all signals for transcriptional initiation as well as transcriptional termination. These LTRs contain both enhancer and promoter regions and are experimentally very well defined. Though retroviral LTRs are homologous in function, LTRs from different groups of retroviruses do not show significant overall sequence homology. Even LTRs from a single group like the Lentiviruses (e.g. HIV, SIV or VISNA) are very different in their overall sequences [14].

3.1 *Generation of models for C-type and Lentivirus LTRs*

We had two well defined training sets of sequences for generation of these models. The training set for the C-type LTRs contained the 17 LTRs that have already been analyzed by CoreSearch [13] plus one additional sequence (musergl1). The Lentivirus LTR training set consisted of 20 different LTRs (more details are described in [14]). The initial model used in both cases was presence of a TATA box followed by a polyA signal (AATAAA). We first analyzed the TATA boxes present in the training sets and established individual consensus profiles for C-type and Lentivirus TATA boxes with the program ConsInd [5]. The derived consensi are remarkably different in details as shown in Figure 1. Especially the region between

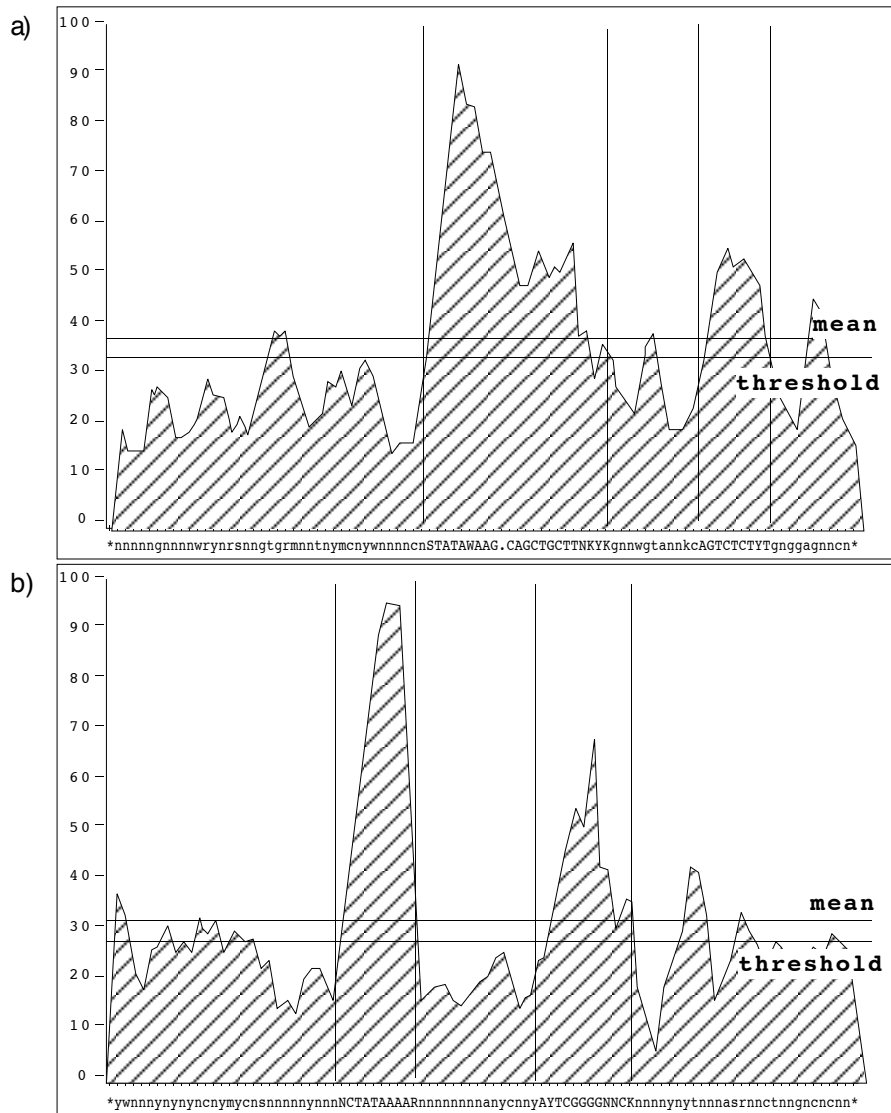


Figure 1: Consensus index profiles. The sequence regions determined to be significant are printed in upper case and marked by two vertical lines. The average consensus index and the threshold used to define the conserved regions are represented by horizontal lines.

a) Consensus index profile of Lentivirus LTR TATA box

b) Consensus index profile of C-type LTR TATA box

the TATA box and the cap-site is much better conserved in Lentivirus LTRs. We suspected these consensi to be specific for the respective LTR type and scored all TATA boxes of both sets against both consensi with ConsInspector [5].

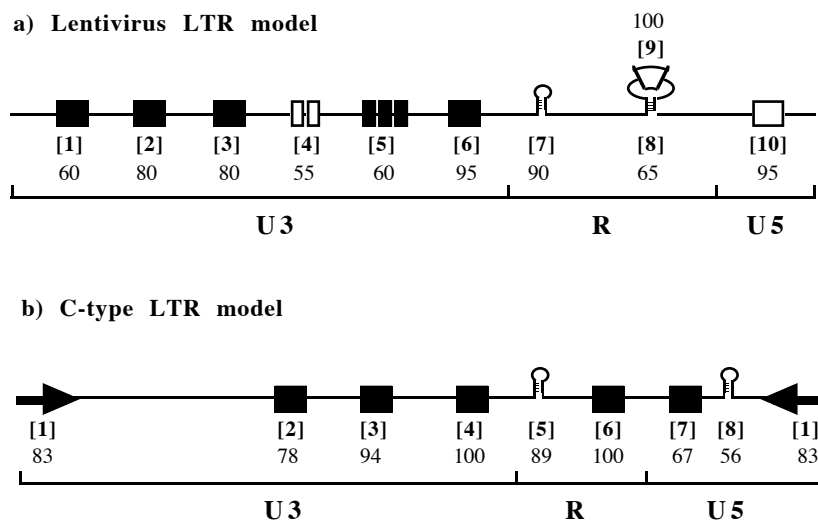


Figure 2: Graphical representations of the Lentivirus and the C-type LTR models

Consensus elements identified by ConsInspector are shown as black boxes. Elements found by MatInspector are shown as white boxes. Hairpins are indicated by a simple hairpin symbol regardless of their true structure. Inverted repeats are shown as black arrows. Bold numbers in brackets [#] identify individual elements and numbers below these identifiers show the frequency of the respective elements (in % of sequences containing the elements).

a) Lentivirus LTR model: [1] primate element block, [2] TATA upstream element, [3] Bel-1 similar region, [4] NF- κ B sites, [5] SP-1 sites, [6] TATA box, [7] TAR region, [8] polyA signal hairpin, [9] polyA signal, [10] polyA downstream element.

b) C-Type LTR model: [1] terminal inverted repeat, [2] upstream signal, [3] CCAAT box, [4] TATA box, [5] R-region hairpin, [6] polyA signal, [7] polyA downstream element, [8] U5 hairpin.

We chose a cutoff score of 50% of the mean consensus score calculated by ConsInd. None of the C-type TATA boxes reached the cutoff score of the Lentivirus TATA box consensus (1.81) while all were above the cutoff score of the C-type TATA box consensus (1.48). However, three Lentivirus TATA boxes scored above C-type cutoff (sivcps, fivltr, pumaltr) and one Lentivirus TATA box did not reach the Lentivirus cutoff score (bimpevpp). This indicated that even the most specific

single element was not sufficient to identify the LTRs. A similar analysis carried out with the polyA signal was even less discriminative for C-type and Lentivirus LTRs (data not shown). However, a combination of both signals yielded better results. There was no TATA/polyA combination in Lentivirus LTRs for which both elements exceeded the thresholds for C-type specific elements and vice versa.

The CoreSearch and ModelGenerator analysis of the LTR training sets revealed additional elements for both LTR types yielding a total of 8 elements for the C-type model (Frech *et al.*, in preparation) and 10 elements for the Lentivirus model [14]. The complete structure of the two LTR models is depicted in Figure 2.

As mentioned in the System and Algorithm section we used distance histograms rather than assuming Gaussian distribution of the distances which would allow calculation of the average value and the standard deviation. The reason for this is shown in Figure 3 which represents the distance histogram derived from the analysis of the distances between the Lentivirus TATA box and the Lentivirus polyA signal. As can be clearly seen there were three preferred distance regions and the average value of the distance distribution (129 bp) would have been totally misleading since it occurred only in 2 of 19 sequences.

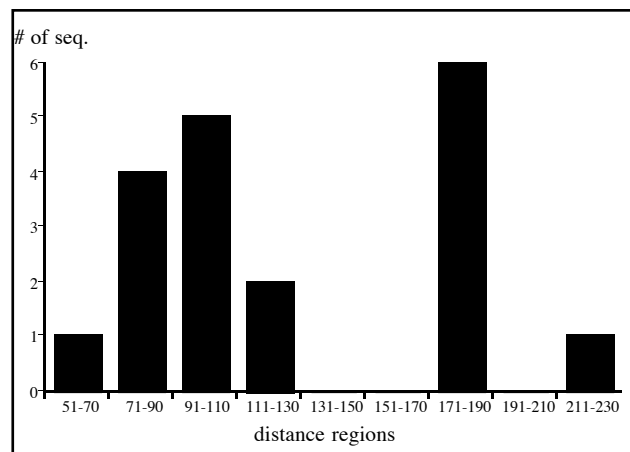


Figure 3: Distance distribution between Lentivirus TATA box and polyA signal
The distance histogram is determined from the training set of Lentivirus LTR sequences. The minimum distance is 64 bp, the maximum distance 214 bp.

3.2 Evaluation of the LTR models

Both LTR models developed by ModelGenerator were used with the program ModelInspector in order to assess the specificity of the models. We already showed that each of the 20 Lentivirus LTR sequences is recognized by a model build from the other 19 sequences [14]. First we analyzed which LTRs were recognized by each model. For that purpose we matched all LTRs from both training sets against both models. Table 1 shows the results of this test. Both models fully recognized their respective training set and clearly discriminated against the other LTR type. Moreover, both models were sensitive to the differences of subgroups within their own training set (Table 1). The different scoring ranges correlated with the biologically defined subgroups with only a very small overlap (0.1) in case of primate and non-primate Lentiviruses. This overlap was solely due to sivsyk which lacked three elements present in all other primate Lentiviruses. These results suggested that both LTR models were very specific. However, in order to verify this it was not sufficient to test the models only with the training set. Therefore, we analyzed all mammalian sequences of GenBank Release 92.0 (sections primate, rodent, and other mammalian, altogether 88,029,873 nucleotides in 89,592 sequences) with ModelInspector and the Lentivirus LTR model. In addition the 1994 Release of the Human Retroviruses and AIDS database [15] containing more than 100 Lentivirus sequences including the LTRs was analyzed in order to determine the number of false negative matches.

Table 1: Cross-validation of LTR models

	primate Lentivirus LTRs			non-primate Lentivirus LTRs			C-type LTRs
	min.	max.	mean	min.	max.	mean	
Lentivirus LTR	5.3	8.6	7.5	4.2	5.4	4.8	0
	exogenous C-type LTRs			endogenous C-type LTRs			Lentivirus LTRs
	min.	max.	mean	min.	max.	mean	
C-type LTR	7.2	7.9	7.4	3.8	5.3	4.6	0

ModelInspector correctly identified all known Lentivirus LTRs in the HIV database (identified in the annotations) indicating that the program produced no false negative matches. It is remarkable that the HIV database also included a number of partial LTRs (usually missing U5 region) which were nevertheless correctly identified. However, presence of less than half of the elements impairs detection excluding small LTR fragments. Table 2 shows the average scores for LTRs from the HIV-1, HIV-2, SIV, AGM, MND, SYK, and RELATED Lentiviruses subgroups as determined from the HIV database search. ModelInspector detected 179 new candidate LTRs on both strands in the mammalian sequences of GenBank in addition to the already known LTRs. Even if all of these new matches would be regarded false positives this would be a false positive rate of about 1 / million nucleotides (exact: 1 / 983,574 nucleotides). This GenBank database search took more than a month on a DEC Alpha 3000/600, but the search time can be reduced to less than one hour by a modified approach involving sequence preanalysis (Frech *et al.*, in preparation).

Table 2: ModelInspector ratings (element score) for Lentivirus LTRs from HIV database

	# of LTRs (partial LTRs)		min. score	max. score	mean score
HIV-1	37	(12)	4.5	7.8	6.6
HIV-2	13	(4)	4.8	7.2	6.5
SIV	21	(6)	5.8	8.5	7.6
AGM	17	(2)	5.1	8.1	6.9
MND	1	(1)	6.7	6.7	6.7
SYK	2	(1)	4.6	5.2	4.9
RELATED	16	(2)	3.5	5.4	4.2

4 Discussion

We described a new algorithm to model transcriptional regulatory regions implemented into a program package consisting of the ModelGenerator program to develop the model and the ModelInspector program capable of searching whole databases for matches to predefined models. ModelInspector has no limit for sequence length allowing the analysis of sequences of several million nucleotides in length such as complete yeast chromosomes. A set of at least 10 sequences and a simple initial model are required as input for ModelGenerator. Most of the model is generated by sequence analysis allowing the development of complex models revealing unknown elements which are part of the organizational framework of the final model. We showed this to work for two groups of retroviral LTRs and demonstrated the discriminative power of the models by cross-validation of the training sets as well as by database searches with one of the models.

ModelGenerator and ModelInspector are using a much more detailed model for regulatory sequences than PromoterScan [8], e.g. our methods use matrix-based TF-site searches in contrast to IUPAC searches employed by PromoterScan. This might explain part of the higher specificities. We have also already shown that the C_i -scores assigned by ConsInspector apparently correlate to some extent with biological functionality [16]. Thus, our approach might also directly benefit from the higher specificity of matrix-based element descriptions as compared to IUPAC searches. The promoter models used by FunSiteP [9] account for the element composition of promoters but do not include detailed organizational features as the models generated by ModelGenerator. We decided to use distance histograms as empirical measures derived from the training set because especially short distances are crucially influenced by sterical restrictions like mutual orientation of adjacent elements with respect to the helix axis. This restriction should be less effective for longer distances which is exactly what we observed.

Although LTRs include polymerase II promoters the models involved more elements in addition to the promoter. These additional elements will most likely contribute to the significant increase in specificity. We showed the LTR models to produce no false negative matches (as far as could be tested). We obtained some preliminary data in the meantime indicating that at least one LTR candidate extracted by ModelInspector from the database has promoter activity which is one of the hallmarks of functional LTRs. Therefore, the number of false positives is lower than the number of additional matches, although we could not yet determine to what extent.

ModelGenerator develops the model from a set of sequences and is not restricted to promoters or LTRs. Any detectable organization of elements within a set of

sequences can therefore provide the basis for model development (examples could be enhancers, scaffold attachment regions or even gene models). The ability to construct models for quite different genetic units is one of the main difference to other model based approaches like GeneParser [11], and GRAIL [12] which are special-purpose gene prediction programs. Model generation with GeneLang [10] requires designing a grammar for a completely known model.

A clear advantage of all multi component models over detection of individual elements (e.g. TATA box) is the ability to tolerate missing individual elements without impairing recognition of the sequence by the model. The high specificity of our LTR models suggested that multi component ModelGenerator models might be suitable to distinguish between different promoter classes, e.g. identify promoters responding to a common pathway as a group. Analysis of the promoter regions of glycolytic enzymes in *saccharomyces cerevisiae* [2,3] indicated that this is possible in principle.

The general design of ModelGenerator also allows to use predefined models as elements for the construction of even more complex models. For example, the described LTR models could be used to develop a model for a complete provirus consisting of a 5'-LTR, a leader region, at least three reading frames for the common retroviral proteins (gag, pol, and env) and a 3'-LTR. Alternatively, a promoter model could be combined with a gene model derived by another method like GRAIL, GeneLang or GeneParser to create a more complex gene model. Of course, ModelGenerator promoter models could be implemented into one of the above methods using ModelInspector as search engine as well.

ModelGenerator integrates a variety of methods which allows extension of the basic model without further *a priori* knowledge or the necessity to alter the program. Thus, new information can be derived directly from the final model which may be a useful prospective analysis facilitating experimental design.

Acknowledgements

We thank Kerstin Quandt for critically reading the manuscript. This work was supported in part by the BMBF Verbundprojekt GENUS 413-4001-01 IB 306 D (Förderschwerpunkt Bioinformatik) and by EU grant BI04-CT95-0226 (TRADAT).

References

1. Hanks, M., Wurst, W., Anson-Cartwright, L., Auerbach, A.B., and Joyner, A.L., *Science* **269**, 679 (1995).
2. Quandt, K., Grote, K., and Werner, T., *Genomics* **33**, 301 (1996).
3. Quandt, K., Grote, K., and Werner, T., *Comp. Appl. Biosci.*, in press (1996).
4. Chen, Q.K., Hertz, G.Z., and Stormo, G.D., *Comp. Appl. Biosci.* **11**, 563 (1995).
5. Frech, K., Herrmann, G., and Werner, T., *Nucleic Acids Res.* **21**, 1655 (1993).
6. Prestridge, D.S. and Stormo, G., *Comp. Appl. Biosci.* **9**, 113 (1993).
7. Quandt, K., Frech, K., Karas, H., Wingender, E., and Werner, T., *Nucleic Acids Res.* **23**, 4878 (1995).
8. Prestridge, D.S., *J. Mol. Biol.* **249**, 923 (1995).
9. Kondrakhin, Y.V., Kel, A.E., Kolchanov, N.A., Romashchenko, A.G., and Milanesi, L., *Comp. Appl. Biosci.* **11**, 477 (1995).
10. Dong, S. and Searls, D.B., *Genomics* **23**, 540 (1994).
11. Snyder, E.E. and Stormo, G.D., *J. Mol. Biol.* **248**, 1 (1995).
12. Uberbacher, E.C. and Mural, R.J., *Proc. Natl. Acad. Sci. USA* **88**, 11261 (1991).
13. Wolfertstetter, F., Frech, K., Herrmann, G., and Werner, T., *Comp. Appl. Biosci.* **12**, 71 (1996).
14. Frech, K., Brack-Werner, R., and Werner, T., *Virology*, in press (1996).
15. Myers, G., Wain-Hobson, S., Henderson, L.E., Korber, B., Jeang, K.-T., and Pavlakis, G.N., Database by Los Alamos National Laboratory (1994).
16. Quandt, K., Frech, K., Herrmann, K., and Werner, T. in *Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism* (Eds. D. Schomburg; U. Lessel), 47, (1995).