# Deliverable 1.2:

# Extending the AMASS Platform via Feature Encoders

by

Juan Maria Sanchez (VCR)
Pal Rujan (LCI)

## 1. Introduction

The main promise of the AMASS platform is that it can be easily extended to other application domains by redefining the way we encode relevant information into the SAMs (Signature Attribute Matrices). Note that LCI's C;A;R;E; (Content Addressable Record Extraction) library itself is built around the concept of records consisting of independent fields. In a more abstract way, it implements a set of strings, where the strings are one-dimensional sequences like person or company names, phone numbers, etc. Version 2.0 will generalize this to either a sequence of sequences (needed in natural text analysis) or to a tree of strings. These extensions in scope are needed by specific applications requiring different types of correlations between the fields.

In this paper we consider two possible applications based on the "high-level"= software CARE API. The first one is demonstrating how to deal with very large streams of data. For the sake of simplicity we consider one example taken from bioinformatics, more specifically searching protein databases. While there exists several standard search algorithm for identifying exact or similar proteomic sub-sequences (see http://srs.ebi.ac.uk/srsbin/cgi-bin/wgetz), they are relatively slow compared with the Sequencer-based version.

The second application deals with how to encode image features into CARE, so that the search based on the image content (Content Based Image Retrieval =CBIR) could use the same framework.

## 2. Application for Content-Based Image Retrieval (CBIR)

The concept of "fuzzy" search engine provided by CARE, plus its hardware acceleration through the developments achieved in the AMASS project, is very well suited for Content-Based Image Retrieval (CBIR) applications. In this kind of applications, we have a usually large database of images, and, given a sample query image, we would like to obtain the images from the database that are more similar to our query image from a "visual" point of view. The concepts of "fuzzy matching" and "result scoring" are a must, given that there are many factors that can introduce variations between two images, such as lightning conditions, pose, signal noise or gamma, while the visual similarity is kept. An exact search would be extremely ineffective for image retrieval.

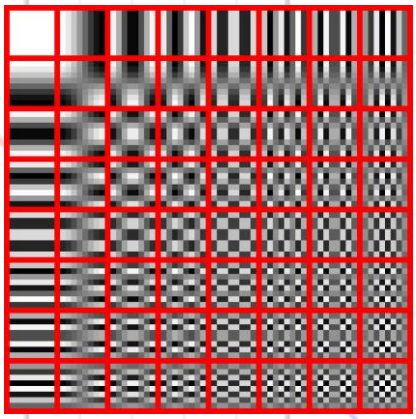
*Two close-up shots of blonde, blue-eyed, smiling women. Visual similarity is obvious, while pixel values are different.*

The representation of the image contents is given by one or more image feature descriptors. When selecting proper descriptors, we must take into account that we can represent an image in a local or a global basis. Feature descriptors can be extracted for image regions (for instance non-overlapping square regions of a certain size), for the image as a whole, or as statistics over image regions descriptors (for instance average and standard deviation of the color histograms for each 16x16 image region). The choice of the right descriptors or image features depends also on a more precise definition of the task: what kind of similarity we are looking for. For example, other descriptors are needed for 'finding smiling persons' than 'find smiling woman with blond hair'. However, in CBIR approaches the system is presented with an image-query and usually no additional information about the task is possible.

In a "fuzzy" framework, we can assume that typical image variations due to lightning, noise, pose, etc. are absorbed by the search engine, so we could initially apply any kind of feature descriptor without worrying about its own invariance symmetries.

## 2.1 The DCT2 Feature Descriptor

There are many feature descriptors that have been developed so far and can be applied for CBIR applications. A frequently used descriptor are the 2D Discrete Cosine Transform (DCT2) coefficients, which represents the frequency structure of the contents and can deal with color information if we take DCT2 coefficients for the different image planes, depending on image pixel representation (RGB, HSV, YCbCr…) One important advantage of using DCT2 as image feature descriptor is that most current image and video compression methods are based on it (MPEG-1, MPEG-2, JPEG…), so that obtaining this descriptor from compressed media implies low-cost partial decompression of the contents. In compressed media, DCT2 is usually applied for 16x16 or 8x8 pixel blocks and on the YCbCr color space.



*2D DCT frequencies*

We can create a global image descriptor based on the local DCT2 coefficients, for instance taking their average and standard deviation for each frequency. We can also concatenate the vectors obtained for the three color components (actually one luminance component Y, and two chrominances Cb and Cr).

## 2.2 Descriptor coding

In order to be used with CARE, the descriptor must be encoded in such a way that it can be understood just like text data. Let us assume that we will use the DCT2 global image descriptor that we have defined in the previous section based on 16x16 non-overlapping regions. We would obtain a representation by a vector of length 16x16x2x3 = 1536 elements. 16x16 is the size of the block, and therefore the number of DCT2 coefficients. Taking average and standard deviation makes it times 2, and concatenating for all three YCbCr components makes it times 3.

If we represent each element as a 16-bit integer value through the proper transformations, we can understand the representation as a very long word encoded using UTF-16. In case this word is too long, we can drop some of the higher-frequency DCT2 coefficients, taking into account that this will cause a

certain amount of smoothing on the image. For instance, if we would like to work on 128-element vector representations, we could use the first 21 DCT2 coefficients (following a zig-zag ordering), and we will obtain vectors of length 21x2x3 = 126 elements.

Once we have encoded the "words" that represent our images, we can use CARE fuzzy search capabilities to cope with variations on this representation and obtain ranked results based on image similarity.

## 3. Searching Streams: Find Similar Protein Sub-sequences

One of the most often encountered search problems in bioinformatics is to identify similar sub-sequences to a given DNA or protein sequence from a database containing all known sequences for a given organism. Typically, DNA and RNA databases use 4 symbols {**A, C, T, G**} or sometimes an additional **N** for a not-identified base. Protein sequences use 21 letters corresponding to the 21 amino-acids. While for genomic sequences we must recode the alphabet into codons (groups of successive three bases taken in a given direction and frame of the DNA sequence), this is not necessary for the protein sequences and one can apply the hardware support directly.

In order to get a feeling of how such a database looks like, we copy below a simple protein of the baker's yeast (saccharomyces cerevisiae) database:

Header:

>NR_SC:SW-PABP_YEAST SW:PABP_YEAST P04147 saccharomyces cerevisiae (baker's yeast). polyadenylate-binding protein, cytoplasmic and nuclear (pabp) (ars consensus binding protein acbp-67) (polyadenylate tail-...

Followed by the protein sequence itself:

ADITDKTAEQLENLNIQDDQKQAATGSESQSVENSSASLYVGDLEPSVSEAHLYDIFSP
IGSVSSIRVCRDAITKTSLGYAYVNFNDHEAGRKAIEQLNYTPIKGRLCRIMWSQRDPSL
RKKGSGNIFIKNLHPDIDNKALYDTFSVFGDILSSKIATDENGKSKGFGFVHFEEEGAAK
EAIDALNGMLLNGQEIYVAPHLSRKERDSQLEETKAHYTNLYVKNINSETTDEQFQELFA
KFGPIVSASLEKDADGKLKGFGFVNYEKHEDAVKAVEALNDSELNGEKLYVGRAQKKNE
RMHVLKKQYEAYRLEKMAKYQGVNLFVKNLDDSVDDEKLEEEFAPYGTITSAKVMRTEN
GKSKGFGFVCFSTPEEATKAITEKNQQIVAGKPLYVAIAQRKDVRRSQLAQQIQARNQM
RYQQATAAAAAAAAGMPGQFMPPMFYGVMPPRGVPFNGPNPQQMNPMGGMPKNGMP
PQFRNGPVYGVPPQGGFPRNANDNNQFYQQKQRQALGEQLYKKVSAKTSNEEAAGKIT
GMILDLPPQEVFPLLESDELFEQHYKEASAAYESFKKEQEQQTEQA

Such databases are in the public domain and accessible either at the NIH (http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide) or in Europe at http://www.ebi.ac.uk/embl/ .

In a typical situation biologists use many queries of a fixed length (20-30 nucleotides long) called markers in order to identify parts of one or more genes

or proteins and their mutations. Assuming the maximal search length is N symbols, it is easy to show that taking windows of 2N-1 length and then shifting them by N symbols will cover the whole sequence in such a way as to find exact matches, should they exist. For simplicity, we use a file of 5,7 MB ASCII text as input. Hence, our "words" are now defined as the contents of these windows and we associate with them two keys, one file pointer to the description of the protein and a second one to the offset of the word-window. The loading, windowing, coding, and storage of the SAMs take about 1,6 sec. When we then perform as example the following query:

- Enter to test query ELDQRGRIIAEYVWI (full)...
  Search time is 33ms on Pentium 2.2 GHz, 0.8 ms on FPGA

  Results:

  Rank = 1 KEY = 3622
  >NR_SC:SW-GLNA_YEAST SW:GLNA_YEAST P32288 saccharomyces cerevisiae (baker's yeast). glutamine synthetase (ec 6.3.1.2) (glutamate--ammonia ligase). 12/98
  Q: ELDQRGRIIAEYVWI
  T: AEASIEKTQILQKYLELDQRGRIIAEYVWI Conf = 100.000000 (similarity)
  match key = 3776 (we return the window content and the file pointer to the protein sequence).

  Rank = 2 KEY = 200161
  >NR_SC:GP-AAA34644_1 gi|171598|gb|AAA34644.1 (M65157) glutamine synthetase [Saccharomyces cerevisiae]
  Q: ELDQRGRIIAEYVWI
  T: ELDQRGRIIAEYVWIDGTGNLRSKGRTLGH Conf = 100.000000
  match key = 200280

  Rank = 3 KEY = 200622
  >NR_SC:PIR-S61058 PIR:S61058 glutamate--ammonia ligase (EC 6.3.1.2) - yeast (Saccharomyces cerevisiae); gi|1314109|emb|CAA94985.1 (Z71255) Gln1p [Saccharomyces cerevisiae]; gi|1072403|emb|CAA92141.1 (Z68111
  Q: ELDQRGRIIAEYVWI
  T: ELDQRGRIIAEYVWIDGTGNLRSKGRTLKK Conf = 100.000000
  match key = 200846

  Rank = 4 KEY = 201214
  >NR_SC:GP-CAA89289_1 gi|809600|emb|CAA89289.1 (Z49274) Gln1p [Saccharomyces cerevisiae]
  Q: ELDQRGRIIAEYVWI
  T: ELDQRGRIIAEYVWIDGTGNLRSKGRTLKK Conf = 100.000000
  match key = 201319

Rank = 5 KEY = 200161
>NR_SC:GP-AAA34644_1 gi|171598|gb|AAA34644.1 (M65157) glutamine synthetase [Saccharomyces cerevisiae]
Q: ELDQRGRIIAEYVWI
T: MAEASIEKTQILQKYLELDQRGRIIAEYVW Conf = 96.000000
match key = 200264

Rank = 6 KEY = 200622
>NR_SC:PIR-S61058 PIR:S61058 glutamate--ammonia ligase (EC 6.3.1.2) - yeast (Saccharomyces cerevisiae); gi|1314109|emb|CAA94985.1 (Z71255) Gln1p [Saccharomyces cerevisiae]; gi|1072403|emb|CAA92141.1 (Z68111
Q: ELDQRGRIIAEYVWI
T: MAEASIEKTQILQKYLELDQRGRIIAEYVW Conf = 96.000000
match key = 200830

Rank = 7 KEY = 201214
>NR_SC:GP-CAA89289_1 gi|809600|emb|CAA89289.1 (Z49274) Gln1p [Saccharomyces cerevisiae]
Q: ELDQRGRIIAEYVWI
T: MAEASIEKTQILQKYLELDQRGRIIAEYVW Conf = 96.000000
match key = 201303

Note that these are the results of the filtering process via the Sequencer algorithm. We can still have false matches (due to the lossy encoder) but do not allow for losing any single relevant match. In a following step (not implemented yet), a full evaluation of the similarity degree as defined in genetics is still necessary. However, the filtering procedure as implemented in AMASS removes very fast almost all false candidates for this last and computational costly step. We performed some qualitative runtime comparisons with the standard tools provided by the public domain databases (basically FASTA and BLAST-variants) and - as far as the displayed times on the web-sites are realistic - observed a relevant speed increase when using the AMASS platform. This could be highly relevant when noting that the fastest super-computers are used today (Blue-Gene, for example) exactly for solving this type of tasks.

These two examples show that it is not very difficult to create and then map feature descriptors into UNICODE-like words or directly into binary features. The bioinformatics example represents also a good illustration of how one would handle arbitrary streams, like audio streams, for example. However, since we are dealing with fuzzy search, the main effort goes into constructing an appropriate representation, so that only the principal features of the stream – but not the small details – are encoded.

In other cases, like for example the iris recognition problem (the only really secure biometrics property besides genetic analysis), it is possible to construct such descriptors using the phase of the wavelet transforms and generating a 2048 bit descriptor. This descriptor is already binary but longer than the allowed

128 bits the AMASS hardware supports now. However, we could split the 2048 bits into binary words of 128 bits, as in the protein problem above. Since 128 bits contain already very specific information, we could then join the results to see if the different matching pieces correspond to one particular iris instance. Another possibility is simply to reprogram the FPGA for this problem and use a simple Hamming-distance parallel adder (also implemented in software) instead of the Sequencer.

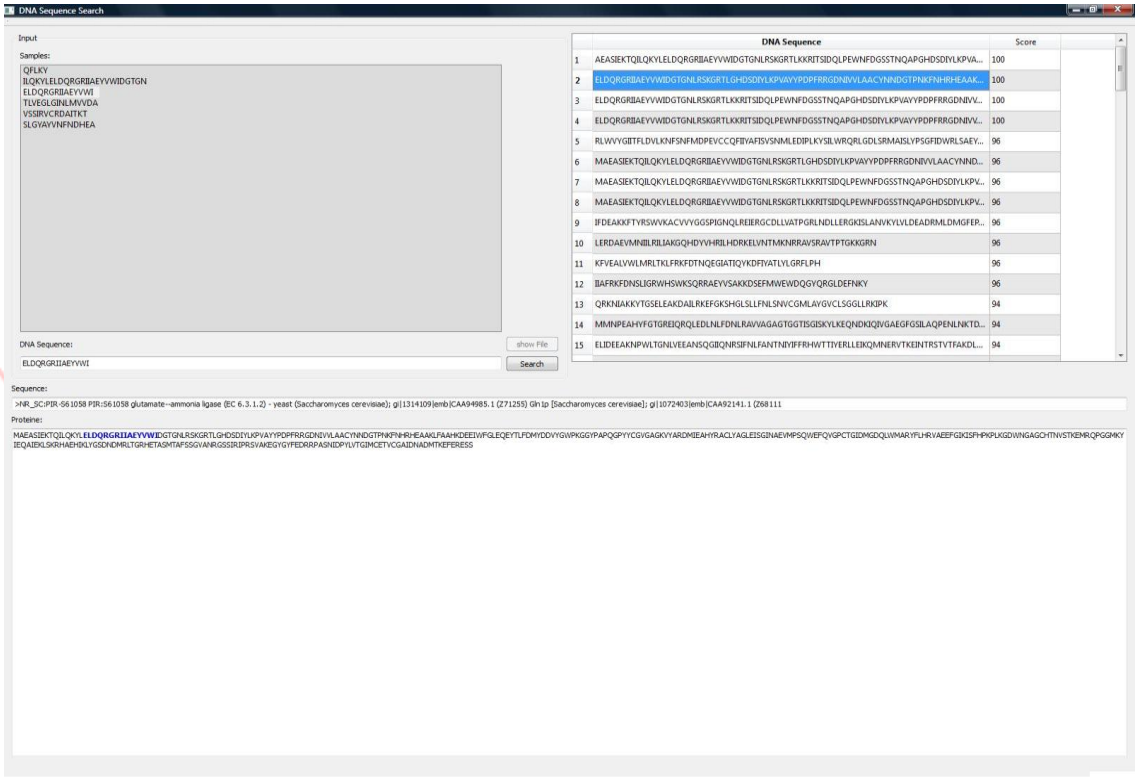
# 4 Conclusions

We have shown above at different levels of concreteness how to extend the number of applications where the AMASS and C;A;R;E; platforms could be used to perform a general approximate search using appropriately designed descriptors or features. The main effort to be invested in the future work is in the following directions:

a) Expand the similarity measures to other quantities than sequences (strings). Good examples are numbers or strongly correlated strings, like when negating a statement.
b) Make sure that C;A;R;E; is scalable to the sizes necessary to handle such large data sets as generated by audio streams or images. In this project we restricted ourselves to a dictionary like approach for natural language words. Here we can safely assume that about 2 million entries would cover almost any language and data corpus. This might be not the case for other problems, including the ones mentioned in this paper. In such cases one must move from the SSRAM to DDR2 or DDR3 dynamic memories, which will slow down somewhat the hardware.
c) A fundamental understanding of the task and a good mathematical background is needed in order to devise appropriate features for approximate search. On this account, the approaches taken here might point to a common way of thinking when solving search problems with pattern recognition methods.

# APPENDIX A

Two snapshots taken from the protein search demonstrator. Please zoom up the images to see details. Please read 'Protein sequence' instead the 'DNA Sequence' (...).


DNA Sequence Search
Input
Samples:
QFLKY
ILQKYLELDQRGRIIAEYVWIDGTGN
ELDQRGRIIAEYVWI
TLVEGLGINLMVVDA
VSSIRVCRDAITKT
SLGYAYVNFNDHEA
DNA Sequence:
ELDQRGRIIAEYVWI
show File
Search
DNA Sequence
Score
1 AEASIEKTQILQKYLELDQRGRIIAEYVWIDGTGNLRSKGRTLKKRITSIDQLPEWNFDGSSTNQAPGHDSDIYLKPVA... 100
2 ELDQRGRIIAEYVWIDGTGNLRSKGRTLGHDSDIYLKPVAYYPDPFRRGDNIVVLAACYNINDGTPNKFNHRHEAAK... 100
3 ELDQRGRIIAEYVWIDGTGNLRSKGRTLKKRITSIDQLPEWNFDGSSTNQAPGHDSDIYLKPVAYYPDPFRRGDNIVV... 100
4 ELDQRGRIIAEYVWIDGTGNLRSKGRTLKKRITSIDQLPEWNFDGSSTNQAPGHDSDIYLKPVAYYPDPFRRGDNIVV... 100
5 RLVVYGIITFLDVLKNFSNFMDPEVCCQFIIYAFISVSNMLEDIPLKYSILWRQRLGDLSRMAISLYPSGFIDWRLSAEY... 96
6 MAEASIEKTQILQKYLELDQRGRIIAEYVWIDGTGNLRSKGRTLGHDSDIYLKPVAYYPDPFRRGDNIVVLAACYNND... 96
7 MAEASIEKTQILQKYLELDQRGRIIAEYVWIDGTGNLRSKGRTLKKRITSIDQLPEWNFDGSSTNQAPGHDSDIYLKPV... 96
8 MAEASIEKTQILQKYLELDQRGRIIAEYVWIDGTGNLRSKGRTLKKRITSIDQLPEWNFDGSSTNQAPGHDSDIYLKPV... 96
9 IFDEAKKFTYRSWVKACVVYGGSPIGNQLREIERGCDLLVATPGRLNDLLERGKISLANVKYLVLDEADRMLDMGFEP... 96
10 LERDAEVMNIILRILAKGQHDYVHRILHDRKELVNTMKNRRAVSRAVTPTGKKGRN 96
11 KFVEALVWLMRLTKLFRKFDTNQEGIATIQYKDFIYATLYLGRFLPH 96
12 IIAFRKFDNSLIGRWHSWKSQRRAEYVSAKKDSEFMWEWDQGYQRGLDEFNKY 96
13 QRKNIAKKYTGSELEAKDAILRKEFGKSHGLSLLFNLSNVCGMLAVGVCLSGGLLRKIPK 94
14 MMNPEAHYFGTGREIQRQLEDLNLFDNLRAVVAGAGTGGTISGISKYLKEQNDKIQIVGAEGFGSILAQPENLNKTD... 94
15 ELIDEEAKNPWLTGNLVEEANSQGIIQNRSIFNLFANTNIYIFFRHWTTIVERLLEIKQMNERVTKEINTRSTVTFAKDL... 94
Sequence:
>NR_SC:PIR:S61058 PIR:S61058 glutamate--ammonia ligase (EC 6.3.1.2) - yeast (Saccharomyces cerevisiae); gi|1314109|emb|CAA94985.1 (Z71255) Gln1p [Saccharomyces cerevisiae]; gi|1072403|emb|CAA92141.1 (Z68111
Proteine:
MAEASIEKTQILQKYLELDQRGRIIAEYVWIDGTGNLRSKGRTLGHDSDIYLKPVAYYPDPFRRGDNIVVLAACYNNDGTPNKFNHRHEAAKLFAAHKDEEIWFGLEQEYTLFDMYDDVYGWPKGGYPAPQGPYYCGVGAGKVYARDMIEAHYRACLYAGLEISGINAEVMPSQWEFQVGPCTGIDMGDQLWMARYFLHRVAEEFGIKISFHPKPLKGDWNGAGCHTNVSTKEMRQPGGMKYIEQAIEKLSKRHAEHIKLYGSDNDMRLTGRHETASMTAFSSGVANRGSSIRIPRSVAKEGYGYFEDRRPASNIDPYLVTGIMCETVCGAIDNADMTKEFERESS
DNA Sequence Search
Input
Samples:
QFLKY
ILQKYLELDQRGRIIAEYVWIDGTGN
ELDQRGRIIAEYVWI
TLVEGLGINLMVVDA
VSSIRVCRDAITKT
SLGYAYVNFNDHEA
DNA Sequence:
TLVEGLGINLMVVDA
show File
Search
DNA Sequence
Score
1 DNGVLRLNEAANVKKTLVEGLGINLMVVDASEEFLSKLKGVTDPEKKRKIIGNTFIHVFEREAEKIKPRDGKEIQFLLQG... 100
2 NGVLRLNEAANVKKTLVEGLGINLMVVDASEEFLSKLKGVTDPEKKRKIIGNTFIHVFEREAEKIKPKDGKEIQFLLQGT... 100
3 DNGVLRLNEAANVKKTLVEGLGINLMVVDASEEFLSKLKGVTDPEKKRKIIGNTFIHVFEREAEKIKPKDGKEIQFLLQG... 100
4 VLATDGNAKALSYDHKPTLASEKSRIVAADGFVEMDRVNGNLALSRAIGDFEFKSNPKLGPEEQIVTCVPDILEHSLD... 96
5 IDAIGKARQKGNFSGANDERENTLNQMLVEMDGFTPADHVVVLAGTNRPDILDKALLRPGRFDRHINIDKPELEGR... 96
6 TVLFFSRDAMFLSTLRISDSGNGIIDDDDEEEYQETHYKSKVLQTAVNFAVIPFWIGFPLSIGLIAWQYRNINAYFITLPF... 96
7 TPEQISSMVLGKMKETAESYLGAKVNDAVVTVPAYFNDSQRQATKDAGTIAGLNVLRIINEPTAAAIAYGLDKKGKEE... 96
8 PIYNQISIQFADLHDRSTRMLVKGVIRNELEWKKSRRFLYWRLRRRLNEGQVIKRLQKKTCDNKTKMKYDDLLKIVQS... 96
9 DAFEAREDYALLKSLFRKKNTVEGISRKMSQESRRLMKAMSTKHSTFQPGWAPRECIENISHLKECIEVKRLDIDDYFL... 96
10 AGLLEPGEVGISATNRNFKGRMGSKDALAYLASPAVVAASAVLGKISSPAEVLSTSEIPFSGVKTEIIENPVVEEEVNAQT... 96
11 DLKPENLLYIDKSDESPLVVADFGIAKTLKSDEELLYKAGTALGYVAPEVLTQDGHGKPCDIWSIGVITYTLLCGYATIDR... 96
12 PEQISSMVLGKMKETAESYLGAKVNDAVVTVPAYFNDSQRQATKDAGTIAGLNVLRIINEPTAAAIAYGLDKKGKEE... 96
13 VIVATAGSTANYKPLTNIVIGEADIKHINVSKVIPQIGPILTILGSALTAPNSSLDDIFGAMKNIADTPALTPLLTLLSNAD... 96
14 LLAADLGGTNFRICSVNILHGDHTFSMEQMKSKIPDDLLDDENVTSDDLFGFLARRTLAFMKKYHPDELAKGKDAK... 96
15 QVSGLETDAPLFRIGNRLVEGSWERLVGTELAFPNAAHVHKKTAGLLSPTEENETTNAGQSKGSSTANDPNIQIQEE... 96
Sequence:
>NR_SC:SW-TFC4_YEAST SW:TFC4_YEAST P33339 saccharomyces cerevisiae (baker's yeast). transcription factor tau 131 kd subunit (tfiic 131 kd subunit). 12/98; PIR:A47453 transcription factor IIIC chain TFC4 -
Proteine:
MAAGEQVSNMFDTILVLDFGSQYSHLITRRLREFNIYAEMLPCTQKISELGWTPKGVILSGGPYSVYAEDAPHVDHAIFDLNVPILGICYGMQELAVINGKQVGRGDKREYGPATLKVIDDSNSLFKGMNDSTVWMSHGDKLHGLPTGYKTLATSDNSPYCGIVHETKPIYGIQFHPEVTHSTQGKTLLKNFAVDLCHAKQNWTMENFIDTEINRIRKLVGPTAEVIGAVSGGVDSTVASKLMTEAIGDRFHALVDNGVLRLNEAANVKKTLVEGLGINLMVVDASEEFLSKLKGVTDPEKKRKIIGNTFIHVFEREAEKIKPRDGKEIQFLLQGTLYPDVIESISFKGPSQTIKTHHNVGGLLENMKLKLIEPLRELFKDEVRHLGELLGIPHDLVWRHPFPGPGIAIRVLGEVTKEQVEIAREADNIYIEEIKKAGLYNQISQAFACLLPVKSVGVMGDQRTYDQVVIALRAIETTDFMTADWAFPFEHSFLKKVASRIVNEVDGVARVTYDITSKPPATVEWE