



(43) International Publication Date
2 August 2012 (02.08.2012)

(10) International Publication Number
WO 2012/101151 A1

(51) International Patent Classification:
G06F 19/20 (2011.01)

(21) International Application Number:
PCT/EP2012/051099

(22) International Filing Date:
25 January 2012 (25.01.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11152232.2 26 January 2011 (26.01.2011) EP

(71) Applicant (for all designated States except US): **AALBORG UNIVERSITET** [DK/DK]; Fredrik Bajers Vej 5, DK-9220 Aalborg (DK).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BUSK, Peter, Kamp** [DK/DK]; Mars Alle 4, DK-2860 Søborg (DK). **LANGE, Lene, M.** [DK/DK]; Karensgeade 5, DK-2500 Valby (DK).

(74) Agent: **ZBM PATENTS APS**; C4 Videnscenter, Krakasvej 17, DK-3400 Hillerød (DK).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,

CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

— with international search report (Art. 21(3))
— with sequence listing part of description (Rule 5.2(a))

(54) Title: A NOVEL METHOD OF PROVIDING A LIBRARY OF N-MERS OR BIOPOLYMERS

(57) Abstract: The present invention relates to a method of providing a library of n-mer sequences, wherein the library is composed of an n-mer sequence. Also the invention concerns a method of providing a library of biopolymer sequences having one or more n-mers in common. Further provided are specific primers and/or probes for i.a. amplification of nucleic acids.



WO 2012/101151 A1

TITLE: A novel method of providing a library of n-mers or biopolymers

FIELD OF THE INVENTION:

The present invention relates to a method of providing a library of n-mer
5 sequences, in particular primers and/or probes, a method of providing a library of
biopolymer sequences involving the use of the n-mers, a method for providing an
extended biopolymer fragment or full length sequence involving use of the primers,
and use of the selected primers and/or probes for *i.a.* amplification of nucleic acids.

10 **BACKGROUND OF THE INVENTION:**

The starting point for the invention here reported on is the rising demand for
more sustainable solutions to important problems of a modern society.

Demand for more biological processes, products and solutions: Post peak oil era
makes it inevitable that the global society at large will have to change from being
15 based on carbon resources from fossils to using renewables. Renewable carbon
resources are primarily made up of plant materials. The conversion of plant materials
(renewable carbon) to substitute the spectrum of useful and needed products (as
energy, plastics, chemicals etc) we currently get from crude oil is in general achieved
by a conversion of the plant biopolymers by the help of microbial enzymes/proteins.
20 This need puts high demands on discovery of enzymes and auxiliary proteins from
microbes, sufficiently diverse and efficient for converting the wide spectrum of different
types of biomass, available globally: from corn stover and wheat straw over sugar cane
bagasse and empty flower bunches of oil palm to municipality waste and agroindustrial
side streams.

25 The complexity of the biomass available put high demands to the microbial
products: Most agricultural products will have to be reserved for feeding the 9 billion
people as well as for feeding the animals for the food chain. The biomass available for
industrial purposes will by and large in future be crop residue/biowaste materials. Such
materials are primarily composed of plant lignocelluloses, a highly recalcitrant structure
30 which needs a host of enzymes for full decomposition. This put even higher demands
on the discovery of new and improved enzymes of microbial origin.

Protein and enzyme discovery can be based on genome sequencing (confined to one
organism at a time and depending on time consuming annotations), activity screening

(requiring cloning and available high through put assays), and searching for novelty through sequence similarity (e.g. a Polymerase Chain Reaction (PCR) based approach).

For decomposition of cellulose and hemicelluloses PCR it is rather simple to
5 construct suitable PCR primers for discovering novel xylanases (e.g. GH10 and GH11)
and discovering novel endoglucanases (e.g. GH45). The 3D protein structure has
through evolution maintained longer stretches of rather highly conserved regions,
suitable for primer construction. However other of the needed types of enzymes for
cellulose decomposition as the cellobiohydrolases or the auxiliary protein belonging to
10 GH61 have either very high sequence variation within each protein family and/or
limited areas of sufficient conservation of sequence similarity.

The motivation for the present invention was a strong wish for more efficient
PCR based discovery methods. The basis for the thinking was a belief in that it should
be possible to construct primers based on further similarities than what was possible
15 from an alignment approach simply nested in the fact that enzymes and other proteins
have been possible to group in protein families (Cazy data base reference,
B.Henrissat), Families which embrace proteins of even very low sequence similarity
but with important similarities in fold and characteristics/activities. Similarly we based
such aspiration on the fact that an *in silico* Blast search could identify a series of
20 proteins which are only distantly related sequence wise but sharing characteristics as
e.g. grouping in the same protein family (Henrissat, B., 1991).

Our hypothesis was that such possible regions suitable for primer construction
could be identified based on bringing forward an advanced level of pattern recognition.
This approach resulted in an invention which with on the one hand its simplicity; and
25 on the other hand its significant valuable advantage achieved, such as in speed.

The resulting invention has for biopolymers such as proteins, RNA and DNA,
been developed to a spectrum of methods, allowing for improved discovery of novel
proteins/peptides (from biological materials as well as from data bases / *in silico*), for
discovering subgrouping of protein families, for identifying micro RNA target sites, for
30 pinpointing important sequence stretches in known and unknown biopolymers.

Methods for creation of degenerated primers are typically based on sequence
alignment (reviewed by (Chakravorty & Vigoreaux 2010)). The design of the primers is
critically dependent on finding the relevant sequences for alignment. These

sequences are selected according to the problem on hand. For example, when looking for new members of a fungal gene family in an *Aspergillus* species it makes sense to limit the alignment to known genes from related *Aspergillus*. However, often the number and divergence of sequences that can be aligned is limited by the ability to
5 perform a correct alignment and by the ability to identify the most conserved sequence motifs in the aligned sequences.

A host of different programs exist for alignment of amino acid sequences (reviewed by (Do & Katoh 2008) and the task can be quite complicated when highly divergent sequences are aligned.

10 Even when alignment is performed correctly there is a need to present the output in a comprehensive way to be able to identify the most conserved motifs when manual identification of conserved motifs is used. Manual inspection of aligned sequences can be confusing when many sequences are included or when gaps are present in the middle of a conserved motif. An alternative is to use software tools such
15 as Green SCPrimer (Jabado et al. 2006) for alignment and identification of conserved sequences. However, alignment remains restricted to sequences with high similarity.

The MEME and related algorithms (Bailey & Elkan 1995; Price et al. 2003) is a bioinformatic tool that can be used for discovery of conserved motifs in protein sequences. The motif length is not fixed and the motifs identified do not have to be 100
20 % identical in the sequences (http://meme.nbcr.net/meme4_5_0/cgi-bin/meme.cgi). This gives a large degree of freedom in the motif search but the drawback is that only a limited number of sequences can be searched (maximum 40 is recommended) and it is assumed that all the input sequences belong to the family and thus should contain the conserved motifs. The method is great for finding sequence motifs where the
25 requirement for the exact sequence of the motif is not absolute. E.g.; native transcription factor binding sites will often be variations of a sequence motif (Busk & Pagès 1998).

The present invention is more suitable for finding sequences that are 100 % identical to the motif as is necessary for degenerated primers. In addition, the
30 requirement for a non-redundant sequence reduces the degree of freedom of the search but allows for larger freedom of input sequences. Thus a large number of sequences can be put into the algorithm and these sequences do not necessarily need to be from the same family as the algorithm will sort the families out.

Glycosyl hydrolases have been classified into families based on sequence alignment and alignment of hydrophobic stretches (Henrissat 1991). However, further classification is required to comprehend the large family 61 of glycosyl hydrolases (gh61) but this problem has not been solved previously (Harris et al. 2010;

5 Karkehabadi et al. 2008). The present invention can be used to generate efficient PCR primers for the gh61 protein family and to classify the gh61s into 13 subfamilies.

Various methods have been developed as alternatives to sequence alignment. E.g.; chaos game representation creates a picture based on the biological sequences and pictures representing different sequences can be compared (Jeffrey 1990). An
10 important limitation of chaos game representation is that the method is only able to accommodate four different words. This makes the method suitable for nucleotide sequence comparison but difficult to adapt to protein sequences made up of 20 different words/amino acids (Davies et al. 2008; Deschavanne & Tufféry 2008). Another alignment-independent approach for sequence comparison is to count the
15 frequency of all words of a certain length (for example trimers) in each sequence and classify the sequences according to word frequencies (Blaisdell 1986; D'Auria et al. 2006; Cheng et al. 2005; Tomović et al. 2006). Variations of this method include dividing the sequences in subsequences with different chemical properties (for example hydrophilic and hydrophobic) (Strope & Moriyama 2007). These alignment-
20 independent methods require less computation than alignment and can be used for comparison of distantly-related sequences (Vinga & J. Almeida 2003) but do not produce the precise and easily comprehensive overview of sequence similarity and differences that are the hallmark of successful alignment (Arakawa et al. 2009; J. S. Almeida & Vinga 2009; Deschavanne & Tufféry 2008).

25 Word frequency methods for alignment-independent sequence comparison are inspired by text analysis methods (reviewed by (Vinga & J. Almeida 2003)). These methods look for short sequences (words) within protein or DNA sequences and count the number of times each word is repeated within the sequences. The similarity between two or more sequences is calculated by comparing the frequency of each
30 word within the sequences (Tomović et al. 2006; D'Auria et al. 2006; Cheng et al. 2005). A sophistication of this approach is to calculate the statistical probability that a word will occur in a training set of sequences to find the words that have the highest probability of being found in the selected sequences but not in randomly chosen

sequences (Vries et al. 2004). The best words are not the sequences that have the highest frequency in the training set, but the words that have the best discriminating power. These words can be used to calculate the likelihood that sequences not included in the training set have the same properties as the training set. The method
5 use word lengths of up to 4 amino acids. E.g.; a set of 4-mer words can be derived from a training set of G-protein Coupled Receptor sequences (GPCR) and used to predict whether other protein sequences are GPCRs (Vries et al. 2004). A problem with this method is that it is dependent on the quality of the underlying models in the training sets. The training sets are defined by application of Hidden Markov Models to
10 curated seed alignments and several parameters such as reliability of alignment and family size affect the selection of training set and hence the outcome of the method.

Families or related proteins often contain short amino acid motifs that are conserved between the family members (Marchler-Bauer et al. 2011). It is assumed that these motifs are conserved because they are functionally or structurally important
15 for the family. Although the motifs are important they will normally only occur once within each protein e.g. a conserved motif forming an active site. Word frequencies methods that calculate n-mer frequencies within sequences are not designed to find such unique amino acid motifs that are conserved between sequences.

20 **References.**

- Bailey, T.L. & Elkan, C., 1995. The value of prior knowledge in discovering motifs with MEME. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 3, 21-29.
- 25 Busk, P.K. & Pagès, M., 1998. Regulation of abscisic acid-induced transcription. *Plant Molecular Biology*, 37(3), 425-435.
- Chakravorty, S. & Vigoreaux, J.O., 2010. Amplification of orthologous genes using degenerate primers. *Methods in Molecular Biology (Clifton, N.J.)*, 634, 175-185.
- Do, C.B. & Katoh, K., 2008. Protein multiple sequence alignment. *Methods in*
30 *Molecular Biology (Clifton, N.J.)*, 484, 379-413.
- Harris, P.V. et al., 2010. Stimulation of lignocellulosic biomass hydrolysis by proteins of glycoside hydrolase family 61: structure and function of a large, enigmatic family. *Biochemistry*, 49(15), 3305-3316.

Henrissat, B., 1991. A classification of glycosyl hydrolases based on amino acid sequence similarities. *The Biochemical Journal*, 280 (Pt 2), 309-316.

Jabado, O.J. et al., 2006. Greene SCPrimer: a rapid comprehensive tool for designing degenerate primers from multiple sequence alignments. *Nucleic Acids*

5 *Research*, 34(22), 6605-6611.

Karkehabadi, S. et al., 2008. The first structure of a glycoside hydrolase family 61 member, Cel61B from *Hypocrea jecorina*, at 1.6 Å resolution. *Journal of Molecular Biology*, 383(1), 144-154.

Price, A., Ramabhadran, S. & Pevzner, P.A., 2003. Finding subtle motifs by branching
10 from sample strings. *Bioinformatics (Oxford, England)*, 19 Suppl 2, ii149-155.

Clarke, A.J., Drummelsmith, J. & Yaguchi, M., 1997. Identification of the catalytic nucleophile in the cellulase from *Schizophyllum commune* and assignment of the enzyme to Family 5, subtype 5 of the glycosidases. *FEBS Letters*, 414(2), pp.359-361.

15 Lo Leggio, L. & Larsen, S., 2002. The 1.62 Å structure of *Thermoascus aurantiacus* endoglucanase: completing the structural picture of subfamilies in glycoside hydrolase family 5. *FEBS Letters*, 523(1-3), pp.103-108.

Van Petegem, F. et al., 2002. Atomic resolution structure of the major endoglucanase from *Thermoascus aurantiacus*. *Biochemical and Biophysical*
20 *Research Communications*, 296(1), pp.161-166.

Stam, M.R. et al., 2006. Dividing the large glycoside hydrolase family 13 into subfamilies: towards improved functional annotations of alpha-amylase-related proteins. *Protein Engineering, Design & Selection: PEDS*, 19(12), pp.555-562.

Almeida, J.S. & Vinga, S., 2009. Biological sequences as pictures: a generic
25 two dimensional solution for iterated maps. *BMC Bioinformatics*, 10, p.100.

Arakawa, K., Oshita, K. & Tomita, M., 2009. A web server for interactive and zoomable Chaos Game Representation images. *Source Code for Biology and Medicine*, 4, p.6.

Blaisdell, B.E., 1986. A measure of the similarity of sets of sequences not
30 requiring sequence alignment. *Proceedings of the National Academy of Sciences of the United States of America*, 83(14), pp.5155-5159.

Cheng, B.Y.M., Carbonell, J.G. & Klein-Seetharaman, J., 2005. Protein classification based on text document classification techniques. *Proteins*, 58(4), pp.955-970.

D'Auria, G., Pushker, R. & Rodriguez-Valera, F., 2006. IWoCS: analyzing
5 ribosomal intergenic transcribed spacers configuration and taxonomic relationships. *Bioinformatics (Oxford, England)*, 22(5), pp.527-531.

Davies, M.N. et al., 2008. Alignment-Independent Techniques for Protein Classification. *Current Proteomics*, 5, pp.217-223.

Deschavanne, P. & Tufféry, P., 2008. Exploring an alignment free approach for
10 protein classification and structural class prediction. *Biochimie*, 90(4), pp.615-625.

Innis, M.A. et al., 1990. *PCR Protocols: A Guide to Methods and Applications*, Academic Press.

Jeffrey, H.J., 1990. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8), pp.2163-2170.

15 Marchler-Bauer, A. et al., 2011. CDD: a Conserved Domain Database for the functional annotation of proteins. *Nucleic Acids Research*, 39(Database issue), pp.D225-229.

Strope, P.K. & Moriyama, E.N., 2007. Simple alignment-free methods for protein classification: a case study from G-protein-coupled receptors. *Genomics*, 89(5),
20 pp.602-612.

Tomović, A., Jančić, P. & Keselj, V., 2006. n-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer Methods and Programs in Biomedicine*, 81(2), pp.137-153.

Vinga, S. & Almeida, J., 2003. Alignment-free sequence comparison—a review.
25 *Bioinformatics*, 19(4), pp.513 -523.

Vries, J.K. et al., 2004. A sequence alignment-independent method for protein classification. *Applied Bioinformatics*, 3(2-3), pp.137-148.

SUMMARY OF THE INVENTION:

30 The present invention distinguishes itself from all other prior art approaches for grouping and discovery of biopolymers. Prior art can in short be described as follows. Alignment is sequence based one to one comparison, fixed in both direction and spacing. Domain finding is based on sequence recognition of known domain

structures. Blast searches are based on sequence homology and a one to one comparison. The inherent potentials of blast search for finding similarities have not been transformed into a method (e.g. through primer construction) for discovery of novel biopolymers.

5 Therefore, we developed Peptide Pattern Recognition (PPR), which is an alignment-independent method for simultaneous comparison of a large number of highly divergent biological sequences at a time. The core method of PPR consists of two steps: 1. Find a limited number of n-mer short sequences that are highly conserved in a group of longer biological sequences such as proteins or nucleic acid
10 sequences. 2. Select the longer biological sequences that contain more than a threshold number of the n-mer short sequences. Any input sequence that is unrelated to the other input sequences will be discarded. Thus, the output of PPR is a group of related sequences selected from the input and a list of the n-mer short sequences that are most conserved in this group. PPR measures word frequencies as the number of
15 sequences that contain a given n-mer but ignores the number of times that the n-mer occurs within each sequence. This makes PPR fundamentally different from traditional word frequencies methods that measure the number of times a given n-mer occurs within each sequence. PPR is free from bias introduced by training set selection and can be used with any word length (n-mer) and does not depend on removing
20 statistically frequent words from the data set. Another important difference is in the practical outcome of the methods: Vries and coworkers show that their method is able to correctly classify 70 % of unknown proteins but claim that the method can be improved to reach about 85 % (Vries et al. 2004). In practical examples PPR is able to classify proteins with 90 % - 95 % accuracy (according to enzymatic function). This
25 considerably higher accuracy in functional prediction is highly valuable for elucidating the function of unknown proteins, e.g. enzymes of potential industrial use.

The current invention builds on a subdivision of the biopolymer (e.g. DNA, RNA or amino acid sequence) not in individual amino acids or nucleotides and base pairs but in blocks of n-mers. Such subdivision is freely placed in all permutation of each of
30 the biopolymer sequences. Further all subdivisions of all sequences included are compared to each other. Not one to one; and not in any specific sequence. Such multiple n-mer subdivision and multiple comparisons, identity finding and ranking is made possible through proposed algorithm directed program based computerizations.

In its simplest form, the invention can be viewed as a method of generating all possible primers for a group of input sequences and test the primers to find the optimal primers that will identify and provide as many as possible of the input sequences. However, in a more general form the invention generates a number of n-mers that characterize a
5 group of biopolymers.

Such comparisons, identifying level of identity for each of the n-mers allows for ranking in level of identity and for inserting thresholds. Identifying what is unifying, usable for discovery of novelty among nature's own variants, novelty in sequence or novelty in subgrouping; and what is different and excluding.

10 The confirmation of the usefulness and uniqueness of the method of the present invention is possible through molecular biological experiments on biological materials as well as through *in silico* experiments, based on database selection and screenings. The result is surprising both in its simplicity, its speed, efficiency, accuracy and potentials for novel discoveries, opening for creation of real value with regard to
15 improved processes, products and solutions, as well as new biological concepts, insights and understandings, which may open for short cut to even more discoveries, identifying of target sites for novel therapeutics or for novel interactions.

DEFINITIONS:

20 Prior to a discussion of the detailed embodiments of the invention is provided a definition of specific terms related to the main aspects and embodiments of the invention.

The term "n-mer" or "n-mer sequence" as used herein is intended to refer to 2 or more consecutive monomeric units of a biopolymer, which is identified in the
25 biopolymer. The n-mers may be identified from the biopolymer starting from one or the other end of the biopolymer. A biopolymer having N building blocks, wherein N is an integer, will naturally consist of N-1 2-mers, N-2 3-mers, N-3 4-mers and so forth. Thus, for instance, a protein having 100 amino acids will have 99 2-mers, 98 3-mers 97 4-mers and so forth. Some of the n-mers in the biopolymer may be identical.

30 The term "having one or more n-mers in common" as used herein such as in connection with a library of biopolymer sequences means that the provided biopolymer sequences of the library are related in such a way that the group of n-mers identified in the library of biopolymers define and characterize the library, thus all the identified n-

mers will be found in the generated library of biopolymers. After selecting a first library of biopolymers the method may be repeated with a biopolymer from the remaining mixture of biopolymers.

The term “library” as used herein is intended to refer to a well defined group of
5 n-mers (e.g. hexapeptides or nucleic acids of 18 base pairs) or biopolymers, which have been identified and selected from a larger group of n-mers or biopolymers.

The term “environment” as used herein means from nature or a predefined source, e.g. a gene bank, a known micro organism, a mammal (e.g. human), or from an unknown source such as a new micro organism, a pool of unidentified biopolymers,
10 or a mixture of known and unknown sources, an environmental sample e.g. from a mammal or a microorganism.

The term “biopolymer” or “biopolymer sequence” as used herein means a biological molecule, including macromolecules, and molecules produced by a living organism, composed of two or more monomeric subunits, or derivatives thereof, which
15 are linked by a bond or a macromolecule. A biopolymer can be, for example, a polynucleotide or a polypeptide, or derivatives or combinations thereof, for example, a nucleic acid molecule containing a peptide nucleic acid portion or a glycoprotein, respectively. Biopolymers include, but are not limited to, nucleic acids or proteins. Nucleic acids include DNA, RNA, and fragments thereof. Nucleic acids can be derived
20 from genomic DNA, RNA, mitochondrial nucleic acid, chloroplast nucleic acid and other organelles with separate genetic material.

The term “a specified group of biopolymers having a mixture of different biopolymers” as used herein is intended to refer to a group of biopolymers of known or unknown origin or mixtures thereof, which comprises a mixture of different
25 biopolymers, which group is analyzed for the purpose of generating a library.

The term “a specified frequency” as used herein is intended to mean that each individual n-mer as selected is accorded a specific number based on how many biopolymers the n-mer can be identified in. Thus, for instance, a hexapeptide (6-mer)
identified in 7 different biopolymers will be given the frequency 7 and a dipeptide (2-
30 mer) identified in 40 different biopolymers will be given the frequency 40.

The term “predefined source” as used herein is intended to mean a known source, such as a gene bank, data base, a known micro organism, or mixtures thereof.

As used herein, a "polypeptide" refers to a biopolymer that comprises more than about 20 consecutive amino acids. The term "polypeptide" encompasses proteins, fragments of proteins, cleaved forms of proteins, partially digested proteins, and the like, which are greater than about 20 consecutive amino acids.

5 As used herein a "peptide" refers to a biopolymer comprising fewer than about 20 consecutive amino acids.

As used herein, a "polynucleotide" refers to a biopolymer that comprises more than about 100 consecutive nucleotides or modified nucleotides. Polynucleotides include DNA, RNA, m-RNA, r-RNA, t-RNA, cDNA, DNA-RNA duplexes, non-coding
10 RNA etc.

The term "primer" as used herein is intended to refer to a strand of nucleic acid that serves as a starting point for DNA or RNA synthesis.

The term "probe" as used herein is intended to refer to a fragment of nucleic acids or amino acids residues of variable length. The probe is typically a single
15 stranded nucleic acids that may recognize a sequence complementary to the sequence in the probe.

The term "signal sequence" as used herein is intended to refer to a short sequence of amino acid residues, usually at the amino terminus of the nascent polypeptide chain that marks the protein for translocation across a membrane. Such
20 sequences may consist of about 3-60 amino acids residues that direct the transport of a protein. Signal sequences may also be called signal peptides, targeting signals, transit peptides, or localization signals. Signal sequences may consist of one or more subunits. When referring to a polypeptide or protein herein such polypeptide or protein may also include a signal sequence.

25 The term "genetic code" as used herein is intended to refer to set of rules by which information encoded in genetic material (DNA or RNA sequences) is translated into proteins (amino acid sequences) by living cells. The code defines a mapping between tri-nucleotide sequences, called codons, and amino acids. The genetic code consists of 64 triplets of nucleotides. These triplets are called codons. With three
30 exceptions, each codon encodes for one of the 20 amino acids used in the synthesis of proteins. That produces some redundancy in the code: most of the amino acids being encoded by more than one codon. The genetic code is almost universal. The same codons are assigned to the same amino acids and to the same start and stop signals

in the vast majority of genes in animals, plants, and microorganisms. However, some exceptions have been found. Most of these involve assigning one or two of the three STOP codons to an amino acid instead. Moreover, mitochondrial mRNA from animals and certain microorganisms use UGA to encode tryptophan (Trp) rather than as a chain terminator. In addition, mitochondrial mRNA from animals and certain microorganisms use AUA for methionine not isoleucine and all vertebrate mitochondria use AGA and AGG as chain terminators. Yeast mitochondria assign all codons beginning with CU to threonine instead of leucine (which is still encoded by UUA and UUG as it is in cytosolic mRNA).

10 The term "overlapping" as used herein is intended to refer to that the two biopolymer fragments possess sequences in common such that the relative order of linked biopolymers can be assembled. For instance, the two biopolymer fragments have a 100 % identical sequence and that this sequence starts from one end of one biopolymer and from the other end of the other biopolymer in a way that the two
15 biopolymers can be assembled to a longer sequence.

In connection with DNA sequence analysis assembling of contiguous stretches of DNA derived from two or more DNA fragments having contiguous nucleotide sequences into contigs is a well known in the art.

The term "nucleic acid consensus sequence" as used herein is intended to refer
20 to that the sequence is a mixture of sequence that at each position may have several possible nucleic acids since the genetic code is degenerate i.e. more than one codon can specify the same amino acid. Thus, for example each of these 4 nucleic acid codons - CCC, CCG, CCT and CCA - specifies the same amino acid, proline.

The term "degenerate primer" as used herein is used to described mixtures of
25 similar, but not identical nucleotide sequences that if translated will encode the same amino acid sequence in at least one of the six reading frames. For example, the nucleotide sequence corresponding to the amino acid isoleucine might be "ATH", where A stands for adenine, T for thymine, and H for adenine, thymine, or cytosine, according to the genetic code for each codon, using the IUPAC symbols for
30 degenerate bases. Thus the three different sequences ATA, ATT and ATC comprises a host of degenerate nucleotide sequences, where the third position of the sequence is degenerated. One or several of the degenerated positions may be substituted with a modified base that can base pair with one or more natural nucleotides. For example,

inosine, which can base pair with A, C, G and T, may be used instead of A, C, G and T in a degenerated nucleotide sequence used as degenerated primer.

The term "*in silico*" as used herein is an expression used to mean performed on computer or via computer simulation. The phrase was coined in 1989 as an analogy to
5 the latin phrases *in vivo* and *in vitro* which are commonly used in biology and refer to experiments done in living organisms and outside of living organisms, respectively.

The term "suitable distance" as used herein is intended to refer to the distance between the two primers which is necessary to perform a meaningful PCR.

The term "extended biopolymer" as used herein is intended to refer to
10 biopolymer fragments which are extended by alignment of the fragments.

The term "unordered biopolymer fragments" as used herein is intended to mean populations of fragments of biopolymers, such as polynucleotides or polypeptides, that may form part of a larger biopolymer, such as a polynucleotide or a polypeptide fragment, which has not been assembled into such larger biopolymer fragments.

15 The term "desired redundancy" as used herein is intended to mean that the level of degeneracy of a degenerate primer having a specific degree of degeneracy may be altered either by incorporating for example inosine into the primers at positions of three- and four-base degeneracy or to introduce preferential biases in codon usages depending on the tRNA pool of the organism of interest.

20 In molecular biology complementarity is a property of double-stranded nucleic acids such as DNA and RNA as well as DNA:RNA duplexes. Each strand is complementary to the other in that the base pairs between them are non-covalently connected via two or three hydrogen bonds. For DNA, adenosine (A) bases complement thymine (T) bases and vice versa; guanine
25 (G) bases complement cytosine (C) bases and vice versa. With RNA, it is the same except that adenine (A) bases complement uracil (U) bases instead of thymine (T) bases.

Since there is only one complementary base for each of the bases found in DNA and in RNA, one can reconstruct a complementary strand for any single
30 strand. All C bases in one strand will pair with G bases in the complementary strand, etc. This is essential for DNA replication. For example, the complementary strand of the DNA sequence
5' A G T C A T G 3'

is

3' T C A G T A C 5'

The latter sequence is called the reverse complementary strand to the DNA sequence 5' A G T C A T G 3' when it is written with the 5' end on the left and the 3'

5 end on the right

5' C A T G A C T 3'

DETAILED DESCRIPTION OF THE INVENTION:

The present invention relates in a broad aspect to a method of providing a library of n-mer sequences, the method comprising the steps of:

- (i) select one or more biopolymer sequence(s) from a specified group of biopolymers having a mixture of different biopolymers,
- (ii) generate all possible n-mers from the sequence(s) wherein n is an integer of at least 2, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
- (iii) use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s),
- (iv) generate all possible n-mers from the first group, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
- (v) optionally use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a second group of one or more sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences,
- (vi) optionally repeat steps (iv) and (v) until no further biopolymers of the specified group of biopolymers are retrieved, and a definite number of biopolymers are identified, and
- (vii) group all provided n-mers from the biopolymers into the library of n-mer sequences.

In an embodiment hereof further including the step (viii) define specific use of the n-mers of the library.

The library of n-mer sequences may contain one or more n-mers, such as one single n-mer from a 3' untranslated region of an mRNA when looking for a miRNA binding site, or two or more n-mers when one or more biopolymer sequences are analyzed and all possible n-mers are identified. Typically, the library generated will contain a huge number of n-mers which have the common feature that they all originates from the biopolymers used to generate such n-mers, and which biopolymers have been grouped by identification of one or more conserved n-mers being found in the biopolymers. Typically, the specified group of biopolymers having a mixture of different biopolymers are from a predefined source, such as a gene bank, a known organism, sample, protein, gene family, chromosomes from one organism or selection of chromosome sequences or parts hereof from several organisms or from an unknown source such as a new microorganism, a pool of unidentified biopolymers, or a mixture of known and unknown sources, an environmental sample e.g. from a mammal, microorganism, plant sample, mixture of organisms, sample of unordered sequence reads e.g., from one or several organisms, from a database.

When selecting a biopolymer sequence from a specified group of biopolymers one or more biopolymers may be selected, such as one biopolymer which will then be used to generate all possible n-mers from the biopolymer sequence, and such group of n-mers may then be designated to a library. If two or more biopolymer sequences are selected from a specified group of biopolymers then such biopolymers may be known and selected because they belong to the same family of biopolymers, or such biopolymers may not be known. If such biopolymers are not known, they may be identified and selected because they have at least one n-mer in common or the method may be continued with two or more biopolymers in parallel and result in one library, if such biopolymers belong to the same family, or several libraries each of which are identified and classified in individual libraries according to the present invention. A library may comprise one or more libraries according to the n-mers used, and a selected biopolymer used for generation of n-mers may belong to more than one library depending on the n-mers used, for instance, a protein having two different domains may generate one library covering the first domain, based on n-mers

generated from the first domain, and a different library covering the second domain, based on n-mers generated from the second domain.

After identification of the n-mers, the next step is to generate all possible n-mers from the biopolymer sequence(s). All the n-mers may then be used in the next step or
5 optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer. By calculating frequency each n-mer will be assigned an integer based on how many biopolymers that contain the specific n-mer. This then will result in one or more n-mers having a frequency number that indicates whether the n-mer is in the lower or higher
10 end and it can then be ranked accordingly if desired. In a further embodiment the n-mers are ranked according to frequency and only the 90%, such as the 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%, most frequent n-mers are used in step (iii) or (v) or both. In another embodiment the n-mers are ranked according to frequency and only the 2, such as the 5, 10, 20, 50, 100, 200 most frequent n-mers are used in step (iii) or
15 (v) or both. In an additional embodiment these two selection criteria may be combined. In a further embodiment the n-mers are ranked according to frequency and only the 90%, such as the 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%, less frequent n-mers are used in step (iii) or (v) or both. In another embodiment the n-mers are ranked according to frequency and only the 2, such as the 5, 10, 20, 50, 100, 200 less
20 frequent n-mers are used in step (iii) or (v) or both. In an additional embodiment these two selection criteria may be combined. Alternatively, the 3 to 150 n-mers, such as 10 to 120, 20 to 100, 50 to 100, typically 100 n-mers, having the highest frequency are used in step (iii) or (v) or both.

In step (iii) the generated n-mers or a selection of the generated n-mers
25 according to a specified frequency are used to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s), if only one biopolymer is found then it is the same biopolymer as selected from the start, and then this may be used to define a library of n-mers.

30 When two or more biopolymers are found, the next step (iv) is to generate all possible n-mers from the first group, or alternatively as explained above for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer. The identified n-mers

may then be grouped into the library of n-mer sequences, which may then be used for a suitable purpose, such as primers or probes.

If desired, the generated n-mers or a selection of the generated n-mers according to a specified frequency from step (iv) may be used in step (v) to provide a second group of one or more sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences. If no new biopolymers are identified or if desired to stop, then the identified n-mers may then be grouped into the library of n-mer sequences, which may then be used for a suitable purpose, such as primers or probes.

As can be seen it may be desired to stop and select the identified n-mer library before reaching the situation where no further biopolymers and thus no further new n-mers are identified. According to the present method it will always be an option to stop looking for further or new n-mers, and then group all provided n-mers from the biopolymers into the library of n-mer sequences, and optionally define specific use of the n-mers of the library, such as primers or probes.

Alternatively, steps (iv) and (v) may be repeated until no further biopolymers of the specified group of biopolymers are retrieved, and a definite number of biopolymers are identified. Hereafter in the next step all provided n-mers from the biopolymers are grouped into the library of n-mer sequences, and optionally define specific use of the n-mers of the library, such as primers or probes.

In a particular embodiment of the method of the present invention for generating one or more selected primers or probes useful for the detection of at least one sequence from a nucleic acid sample, the method comprises the steps of:
in step (iv) generate all possible n-mers from the first group and for each n-mer sequence calculate the frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
and in step (viii) decide whether the n-mers are peptides or nucleic acids, if the n-mers are peptides reverse translate the n-mers into a corresponding nucleic acid consensus sequence, and provide at least one nucleic acid consensus sequence from one of the most frequent occurring n-mers, and define whether it is a probe or a primer. In a further embodiment the n-mer is a primer. In a still further embodiment the n-mer is a probe.

In a further embodiment of the method of the present invention the nucleic acid consensus sequence of step (iv) is translated according to the genetic code.

In a still further embodiment of the method of the present invention two primers are provided separated by a suitable distance and the reverse primer is
5 complementary to the nucleic acid consensus sequence. Thus the forward and reverse primers can be used for PCR that will generate a sequence comprising the primers and any sequence located between the primers Sambrook, J. & Russell, D.W., 2001. Molecular Cloning: A Laboratory Manual, Third Edition 3rd ed., Cold Spring Harbor Laboratory Press.

10 Typically, the set of primers are selected to be degenerated primers consisting of as few as possible similar, but not identical nucleotide sequences. Thus each degenerated primers may consists of 2, such as 4, 8, 16, 32, 64, 128, 256, 512 1024 similar, but not identical nucleotide sequences.

The n-mers provided according to the above described method of the present
15 invention have many uses such as for amplification of nucleic acids, as hybridization probes for screening a library of nucleotide sequences or an expression library of peptides or polypeptides, as as antigens for generating an antibody for screening of an expression library of peptides or polypeptides, as an expression library of peptides or polypeptides, both in wet lab and *in silico* screening.

20 The above embodiments as well as the embodiments to be described hereunder should be seen as referring to any one of the aspects described herein as well as any one of the embodiments described herein unless it is specified that an embodiment relates to a certain aspect or aspects of the present invention.

In a further aspect the present invention relates to use of primers obtainable
25 from the method of the present invention for amplification of nucleic acids.

In a still further aspect the present invention relates to use of primers obtainable from the method of the present invention as hybridization probes for screening a library of nucleotide sequences

In a still further aspect the present invention relates to use of primers obtainable
30 from the method of the present invention as hybridization probes for screening an expression library of peptides or polypeptides

In a still further aspect the present invention relates to use of primers obtainable from the method of the present invention as antigens for generating an antibody for screening of an expression library of peptides or polypeptides

In a further aspect the present invention relates to use of primers obtainable
5 from the method of the present invention as an expression library of peptides or polypeptides, both in wet lab and *in silico* screening.

In a further embodiment of above uses, the probes used are obtained from the method of the present invention.

In a further aspect the present invention relates to use of probes obtainable
10 from the method of the present invention for amplification of nucleic acids.

In a still further aspect the present invention relates to use of probes obtainable from the method of the present invention as hybridization probes for screening a library of nucleotide sequences

In a further aspect the present invention relates to use of probes obtainable
15 from the method of the present invention as an expression library of peptides or polypeptides, both in wet lab and *in silico* screening.

In a further embodiment of above uses, the probes used are obtained from the method of the present invention.

In a further aspect the present invention relates to a method of providing a
20 library of biopolymer sequences having one or more n-mers in common, wherein the library is composed of at least 2 biopolymer sequences, the method comprising the steps of:

- (i) select one or more biopolymer sequence(s) from a specified group of biopolymers having a mixture of different biopolymers,
- 25 (ii) generate all possible n-mers from the sequence(s) wherein n is an integer of at least 2, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
- (iii) use the generated n-mers or a selection of the generated n-mers according to a
30 specified frequency to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s),

- (iv) generate all possible n-mers from the first group, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
- (v) use the generated n-mers or a selection of the generated n-mers according to a
5 specified frequency to provide a second group of one or more sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences,
- (vi) repeat steps (iv) and (v) until no further biopolymers of the specified group of biopolymers are retrieved, and a definite number of biopolymers are identified,
- (vii) group all provided biopolymers into the library of biopolymer sequences having
10 one or more of the n-mer(s). The steps are performed in a similar manner as explained above in relation to providing a library of n-mers, with the exception that it is now the biopolymers that are the end result.

Accordingly, when a first library of biopolymers is identified and selected, then the method can be repeated on the remaining mixture of different biopolymers, and so
15 forth until all biopolymers from the mixture of different biopolymers have been assigned to a library or have been identified as not belonging to a library. Whether or not all libraries of biopolymers in the mixture of different biopolymers is identified and selected is a matter of choice.

In a further embodiment the biopolymer sequences of the library have sequence
20 similarity of at least 1% identity, such as at least 5%, from 1% to 25%, or from 5% to 50% identity.

Preferred methods to determine identity are designed to give the largest match between the sequences tested. Methods to determine identity are described in publicly available computer programs. Preferred computer program methods to determine
25 identity between two sequences include the GCG program package, including GAP (Devereux et al., *Nucl. Acid. Res.*, 12, 387, (1984)); Genetics Computer Group, University of Wisconsin, Madison, Wis.), BLASTP, BLASTN, and FASTA (Altschul et al., *J. Mol. Biol.*, 215, 403-410, (1990)). The BLASTX program is publicly available from the National Center for Biotechnology Information (NCBI) and other sources
30 (BLAST Manual, Altschul et al. NCB/NLM/NIH Bethesda, Md. 20894; Altschul et al., *supra*). The well known Smith Waterman algorithm may also be used to determine identity.

For example, using the computer algorithm GAP (Genetics Computer Group, University of Wisconsin, Madison, Wis.), two proteins for which the percent sequence identity is to be determined are aligned for optimal matching of their respective amino acids (the "matched span", as determined by the algorithm). A gap opening penalty
5 (which is calculated as 3.times. the average diagonal; the "average diagonal" is the average of the diagonal of the comparison matrix being used; the "diagonal" is the score or number assigned to each perfect amino acid match by the particular comparison matrix) and a gap extension penalty (which is usually {fraction (1/10)} times the gap opening penalty), as well as a comparison matrix such as PAM 250 or
10 BLOSUM 62 are used in conjunction with the algorithm. A standard comparison matrix (see Dayhoff et al., Atlas of Protein Sequence and Structure, vol. 5, suppl. 3, (1978) for the PAM 250 comparison matrix; Henikoff et al., *Proc. Natl. Acad. Sci USA*, 89, 10915-10919, (1992) for the BLOSUM 62 comparison matrix) is also used by the algorithm.

In a still further embodiment the biopolymer sequences of the library have the
15 same bioactivity.

In a further embodiment the biopolymer sequences of the library are from the same gene family.

In a still further embodiment the biopolymer sequences of the library have the same type or phylogenetic class of organisms.

20 In a further embodiment the biopolymer sequences of the library have the same organism.

In a still further embodiment the biopolymer sequences of the library are from the same sample containing biopolymers from one or more organisms.

Once a library of biopolymers have been selected, for instance, because they
25 have the same bioactivity, or maybe are from the same gene family or other desired property, or for further testing to identify the property or properties, such library maybe subjected to the same method of the present invention again, for instance, by subjecting the provided library to steps (i) to (vii) and the n-mer is at least one number higher than the previous defined n-mer.

30 In a further embodiment n is an integer above 1 which may be as high as the number of building blocks in the biopolymers although this may in many cases only provide one or a few biopolymers from the mixture of different biopolymers, typically n is an integer from 2 to 100, such as 2-75, 3-60, 3-18, 4-50, 5-20, 5-8, 2-10, or e.g. 2-6.

In another embodiment the length of the n-mer may be a fraction of the length of the biopolymer such as 1/10, 1/100, 1/1000, 1/10000 or even smaller fraction when the biopolymer consist of very long biopolymers such as chromosomes.

Typically if the biopolymer sequences are selected from polypeptides, e.g. 5 proteins then the n-mer is composed of n amino acids. Moreover if the biopolymer sequences are selected from nucleic acids (e.g. DNA or RNA) then the n-mer is composed of n nucleotides.

In a still further embodiment the biopolymer sequence is selected from polypeptides, such as proteins, or fragments thereof.

10 In a further embodiment the biopolymer sequence is selected from nucleic acids such as polydeoxyribonucleic acids (DNA) and polyribonucleic acids (RNA), or fragments thereof. Typically, when the nucleic acid is an RNA it is selected from mRNA and non-coding RNA, e.g. microRNA.

The present method for providing n-mers or biopolymers may be performed in a 15 lab, on paper or by using a computer, that is, by *in silico*. Typically, in order to generate and handle large data sets the method is performed *in silico*.

In a further aspect the present invention relates to use of an n-mer from a biopolymer sequence having an identified sequence for searching and collecting fragments of biopolymers from an environment comprising the biopolymer fragments.

20 A further use of n-mers generated from a known biopolymer or an identified sequence of a biopolymer is the possibility of looking for biopolymer fragments in an environment where the sequence of such fragments have not yet been identified as belonging to a certain host. By using the n-mers from a biopolymer sequence having an identified sequence it may be possible to search and find biopolymer fragments 25 which have the n-mer in the sequence. Such identified and collected biopolymer fragments may then be assembled to a larger fragment or full length biopolymer, which can then be tested for a specific activity. For instance, from a specific biopolymer sequence n-mers are generated and then such n-mers are used to provide a group of one or more biopolymer fragments from the specified environment, wherein such 30 fragments have one or more of the n-mers in the sequence. If the n-mers are amino acid sequences and the biopolymer fragment from the environment consists of nucleotide sequences then the biopolymer fragment is considered to have the n-mer in the sequence if the biopolymer fragment sequence translated in any of the three

reading frames encodes the n-mer. Likewise, reverse complementary sequence to the biopolymer fragment is considered to have the n-mer in the sequence if the biopolymer fragment reverse complementary sequence translated in any of the three reading frames encodes the n-mer. Hereafter, the collected biopolymer fragments are aligned if possible to generate a longer sequence consisting of overlapping, collected biopolymer fragments. Then repeat, if this is considered suitable, the search for biopolymer fragments by using n-mers generated from the identified biopolymer fragments until the extended biopolymer fragments reach the expected length of the known biopolymer full length or until no new extensions are found.

10 Accordingly the present invention concerns use of a library of n-mer sequence(s), obtainable by the method of providing a library of n-mer sequences of the present invention, from a biopolymer sequence having an identified sequence for searching and collecting fragments of biopolymers from an environment comprising the biopolymer fragments, by *in silico* screening and collecting any biopolymer fragment(s) that has a sequence identical to, complementary to or related by translation or reverse translation to the sequence of one or more of the selected primers and/or probes.

 In a further aspect the present invention relates to use of a library of n-mer sequences generated from a library of one or more known biopolymer sequences having a known function in common to identify a different biopolymer sequence comprising the n-mer sequences and having the same function as the known biopolymer sequence(s), by comparing the library of n-mer sequences of the known biopolymer sequence(s) with the n-mer sequences from the different biopolymer sequence(s), and selecting the different biopolymer sequence(s) having at least 1 of the n-mers in common. Typically, and in order to increase the likelihood that the function of the different biopolymer is the same as the known biopolymer sequence(s) it is preferred to select the different biopolymer sequence(s) having at least 2 of the n-mers in common, such as at least 3 at least 4, at least 5, at least 6, such as 10, such as 20, such as 30 in common. Furthermore, the sum of the frequencies of the common n-mers (frequency is number of biopolymers, e.g. proteins, in the library that contain the n-mer divided by total number of proteins in the library) will typically be used to increase the likelihood that the function of the different biopolymer is the same as the known biopolymer sequence(s), such that the sum of the frequencies of the common

n-mers should be at least 0.5, such as at least 0.8, at least 1.0, at least 1.5, at least 2.0, at least 3.0. However, other functions of the n-mers such as the product of their frequencies may also be used to predict the functions of the different biopolymer sequence.

5 In addition, the function of the different biopolymer sequence may be predicted from a relative comparison of the number of n-mers and/or the sum of their frequencies or other property of the n-mers from one library of n-mers relative to the same function for the n-mers from another library of n-mers.

The different biopolymers to be compared with the library of n-mers from the
10 known biopolymers may be preselected based on sequence identity (e.g. by alignment to the known biopolymers used to generate the library of n-mers), species of origin, expression pattern or other function of the biopolymers.

The known function as used herein is to be understood in its broadest interpretation to mean any function such as and without limitation: sequence identity,
15 species of origin, expression pattern, enzymatic activity, structural role, helper function, epitope, recognition site for proteins, secondary or tertiary structure or any other function that a peptide-, protein-, or nucleic acid sequence may have and which is determined by the sequence of the biopolymer. Each of the specified functions constitutes individual embodiments of the function of the known biopolymer as well as
20 the different biopolymer, and can be combined with any other embodiment of the biopolymer sequences according to the present invention.

In a further embodiment the use further comprises the step of screening the selected different biopolymer sequence(s) in a relevant assay to confirm that the function is the same as the known biopolymer sequence(s).

25 The library of n-mer sequences of the known biopolymer sequence(s) may be compared with the n-mer sequences from the different biopolymer sequence(s) in different ways which are all intended to be encompassed by the present invention without limiting the scope thereof, such as by defining a score of the selected biopolymer for each n-mer library calculated as the number of the n-mers that are
30 included in the selected sequence of the biopolymer or the sum of the frequency of the n-mers that are included in the selected sequence of the biopolymer or as another value associated to the n-mers that are included in the selected sequence of the

biopolymer (e.g. multiplication of the frequencies of the n-mers). The score of each selected biopolymer is used to associate the biopolymer to the library of n-mers and to infer the probability that the selected biopolymer has similar properties as the known biopolymers used to generate the library of n-mers.

- 5 The score is an absolute number, however the score of a biopolymer for one library of n-mers may be compared to the score for another library of n-mers to decide which library (and thus known biopolymers) are most related to the different biopolymer. Even the known biopolymers used to generate the library of n-mers may be scored and found to be closer related to another library of n-mers.

10 List of embodiments

1. A method of providing a library of n-mer sequences, wherein the library is composed of an n-mer sequence, the method comprising the steps of:
 - (i) select a biopolymer sequence from a specified group of biopolymers having a mixture of different biopolymers,
 - 15 (ii) generate all possible n-mers from the sequence(s) wherein n is an integer of at least 2, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
 - (iii) use the generated n-mers or a selection of the generated n-mers according to a
20 specified frequency to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s),
 - (iv) generate all possible n-mers from the first group, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer
25 sequence at least once and assign the frequency to the n-mer,
 - (v) optionally use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a second group of one or more sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences,
 - 30 (vi) optionally repeat steps (iv) and (v) until no further biopolymers of the specified group of biopolymers are retrieved, and a definite number of biopolymers are identified,

(vii) group all provided n-mers from the biopolymers into the library of n-mer sequences, and

(viii) optionally define specific use of the n-mers of the library.

2. The method of embodiment 1 for generating one or more selected primers or probes useful for the detection of at least one sequence from a nucleic acid sample, the method comprising the steps of:

in step (iv) generate all possible n-mers from the first group and for each n-mer sequence calculate the frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,

10 and in step (viii) decide whether the n-mers are peptides or nucleic acids, if the n-mers are peptides reverse translate the n-mers into a corresponding nucleic acid consensus sequence, and provide at least one nucleic acid consensus sequence from one of the most frequent occurring n-mers, and define whether it is a probe or a primer.

3. The method of embodiment 2, wherein two primers are provided separated by a suitable distance and, wherein the reverse primer is complementary to the nucleic acid consensus sequence.

4. The method of any one of embodiments 2-3, wherein the nucleic acid consensus sequence of step (iv) is translated according to the genetic code.

5. The method of any one of embodiments 2-4, wherein primer set further is selected to have a desired redundancy.

6. A method of providing a library of biopolymer sequences having one or more n-mers in common, wherein the library is composed of at least 2 biopolymer sequences, the method comprising the steps of:

25 (i) select a biopolymer sequence from a specified group of biopolymers having a mixture of different biopolymers,

(ii) generate all possible n-mers from the sequence(s) wherein n is an integer of at least 2, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,

30 (iii) use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s),

(iv) generate all possible n-mers from the first group, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,

(v) use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a second group of one or more sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences,

(vi) repeat steps (iv) and (v) until no further biopolymers of the specified group of biopolymers are retrieved, and a definite number of biopolymers are identified,

(vii) group all provided biopolymers into the library of biopolymer sequences having one or more of the n-mer(s).

7. The method of any one of embodiments 1-6 wherein the mixture of different biopolymers are from a predefined source, such as a gene bank, a known micro organism, sample, protein, gene family, chromosomes from one organism or selection of chromosome sequences or parts hereof from several organisms or from an unknown source such as a new micro organism, a pool of unidentified biopolymers, or a mixture of known and unknown sources, an environmental sample e.g. from a mammal, microorganism, plant sample, mixture of organisms, sample of unordered sequence reads e.g., from one or several organisms, or from a database.

8. The method of any one of embodiments 1-7 wherein n is an integer from 2 to 75.

9. The method of any one of embodiments 1-8 wherein the library of biopolymer sequences has sequence similarity of at least 1% identity, or has the same bioactivity, or is from the same gene family, or from the same type or phylogenetic class of organisms, from the same organism, from the same sample containing biopolymers from one or more organisms.

10. The method of any one of embodiments 1-9 wherein the biopolymer sequence is selected from polypeptides, proteins, nucleic acids, or fragments thereof.

11. The method of any one of embodiments 1-10 wherein the n-mers are ranked according to frequency and only the 90% most frequent n-mers are used in step (iii) or (v) or both.

12. The method of any one of embodiments 1-10 wherein 3 to 150 n-mers, such as about 100, having the highest frequency are used in step (iii) or (v) or both.

13. The method of any one of embodiments 1-12 wherein the method is performed *in silico*.

14. Use of the selected primers and/or probes of any one of embodiments 1-13 for amplification of nucleic acids, as hybridization probes for screening a library of
5 nucleotide sequences, as an expression library of peptides or polypeptides, both in wet lab and *in silico* screening.

15. Use of an n-mer from a biopolymer sequence having an identified sequence for searching and collecting fragments of biopolymers from an environment comprising the biopolymer fragments.

10 All references, including publications, patent applications and patents, cited herein are hereby incorporated by reference to the same extent as if each reference was individually and specifically indicated to be incorporated by reference and was set forth in its entirety herein.

All headings and sub-headings are used herein for convenience only and
15 should not be construed as limiting the invention in any way.

Any combination of the above-described elements in all possible variations thereof is encompassed by the invention unless otherwise indicated herein or otherwise clearly contradicted by context.

The terms "a" and "an" and "the" and similar referents as used in the context of
20 de-scribing the invention are to be construed to cover both the singular and the plural, unless otherwise indicated herein or clearly contradicted by context.

Recitation of ranges of values herein are merely intended to serve as a shorthand method of referring individually to each separate value falling within the range, unless otherwise indicated herein, and each separate value is incorporated into
25 the specification as if it were individually recited herein.

Unless otherwise stated, all exact values provided herein are representative of corresponding approximate values (e.g., all exact exemplary values provided with respect to a particular factor or measurement can be considered to also provide a corresponding approximate measurement, modified by "about" where appropriate).

30 All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context.

The use of any and all examples, or exemplary language (e.g., "such as") provided herein, is intended merely to better illuminate the invention and does not pose a limitation on the scope of the invention unless otherwise indicated. No language in the specification should be construed as indicating any element is essential to the
5 practice of the invention unless as much is explicitly stated.

The citation and incorporation of patent documents herein is done for convenience only and does not reflect any view of the validity, patentability and/or enforceability of such patent documents.

10 **EXAMPLES:**

Example 1:

Computer algorithm for classifying related biopolymers in groups and finding n-mer sequences with a predefined frequency and example of classifying animal proteins with related function and sequence in six separate groups.

15

Algorithm

The algorithm to be implemented was:

1. For each biopolymer make all the n-mers that occur in the biopolymer sequence.
- 20 2. Select all biopolymer that contain more than a defined number of the n-mers.
3. Make all the n-mers that occur in these biopolymers and a defined number of the most abundant n-mers.
4. Go back to step 2 until no new n-mers are made in the following round.

25 **Program:**

The program is written in the Ruby programming language version 1.8.6 and normally executed on a machine with the Microsoft Windows XP version 2002 operative system but can also be executed under other operative systems and would easily be adapted to other versions of Ruby.

30

Classify_family3.rb:

```
cut_off = 9  
limit = 100  
35 pep_length = 6
```

```
class Protein
  attr_accessor :seq, :name, :score, :peptides
  def initialize(seq)
    @seq = seq.to_s.upcase.chomp
5  end
  def count_occurrence(x)
    occurrence = 0
    seq = ""
    seq += @seq
10  while seq.include?(x)
      occurrence += 1
      seq = seq.slice(seq.index(x)+1, seq.length)
    end
    occurrence
15  end
  def calc_score(array)
    score = 0
    if array.first.class.to_s == "String"
      array.each{|pep| score += 1 if @seq.include?(pep)}
20  end
    if array.first.class.to_s == "Peptide"
      array.each{|pep| score += 1 if @seq.include?(pep.seq)}
    end
    score
25  end
  def find_family
    #@family_array.sort.last[1]
    array = @family_array.sort.reverse
    if array[0][0] != array[1][0]
30    family = array[0][1]
    else
      family = 0
    end
    family
35  end
  def split_to_20pep
    array = Array.new
    seq = ""
    seq += @seq
40  20.times do
      array << seq.scan(/...../)
      seq.slice!(0,1)
    end
    array.flatten.uniq
45  end
  def split_to_15pep
    array = Array.new
    seq = ""
    seq += @seq
```

```
15.times do
  array << seq.scan(/...../)
  seq.slice!(0,1)
end
5  array.flatten.uniq
end
def split_to_decapep
  array = Array.new
  seq = ""
10  seq += @seq
  10.times do
    array << seq.scan(/...../)
    seq.slice!(0,1)
  end
15  array.flatten.uniq
end
def split_to_nonapep
  array = Array.new
  seq = ""
20  seq += @seq
  9.times do
    array << seq.scan(/...../)
    seq.slice!(0,1)
  end
25  array.flatten.uniq
end
def split_to_octapep
  array = Array.new
  seq = ""
30  seq += @seq
  8.times do
    array << seq.scan(/...../)
    seq.slice!(0,1)
  end
35  array.flatten.uniq
end
def split_to_heptapep
  array = Array.new
  seq = ""
40  seq += @seq
  7.times do
    array << seq.scan(/...../)
    seq.slice!(0,1)
  end
45  array.flatten.uniq
end
def split_to_hexapep
  array = Array.new
  seq = ""
```

```
    seq += @seq
    6.times do
      array << seq.scan(/...../)
      seq.slice!(0,1)
5    end
    array.flatten.uniq
  end
  def split_to_pentapep
    array = Array.new
    10    seq = ""
    seq += @seq
    5.times do
      array << seq.scan(/...../)
      seq.slice!(0,1)
15    end
    array.flatten.uniq
  end
  def split_to_tetrapep
    array = Array.new
    20    seq = ""
    seq += @seq
    4.times do
      array << seq.scan(/..../)
      seq.slice!(0,1)
25    end
    array.flatten.uniq
  end
  def split_to_tri pep
    array = Array.new
    30    seq = ""
    seq += @seq
    3.times do
      array << seq.scan(/.../)
      seq.slice!(0,1)
35    end
    array.flatten.uniq
  end
  def split_to_dipep
    array = Array.new
    40    seq = ""
    seq += @seq
    2.times do
      array << seq.scan(/../)
      seq.slice!(0,1)
45    end
    array.flatten.uniq
  end
  def mark_sequence_stretch(seq1, seq2)
    cut1 = @seq.index(seq1.seq)
```

```

    cut2 = @seq.index(seq2)+seq2.length
    amplicon_length = cut2-cut1
    x = @seq.slice(0, cut1) + "[ "
    y = @seq.slice(cut1, amplicon_length).upcase + " ]"
5    z = @seq.slice(cut2, @seq.length)
    x+y+z
  end
end
class Peptide < Protein
10  attr_accessor :seq, :name, :score, :degeneracy, :degeneracy_w_inosine,
    :average_position, :frequency, :prot_score
    def initialize(seq)
      @seq = seq.to_s
    end
15  def calc_score(array)
    score = 0
    array.each{|p| score += 1 if p.seq.include?(@seq)}
    score
  end
20  def calc_degeneracy
    antal_codons = {'a' => '4', 'c' => '2', 'd' => '2', 'e' => '2',
      'f' => '2', 'g' => '4', 'h' => '2', 'i' => '3', 'k' => '2',
      'l' => '8', 'm' => '1', 'n' => '2', 'p' => '4', 'q' => '2',
      'r' => '8', 's' => '16', 't' => '4', 'v' => '4', 'w' => '1', 'y' => '2'}
25  array = @seq.split(//)
    degeneracy_number = 1
    degeneracy_number = 4 if array.last.downcase == 's'
    degeneracy_number = 2 if array.last.downcase == 'r'
    degeneracy_number = 2 if array.last.downcase == 'l'
30  array.pop
    array.each {|x| degeneracy_number *= antal_codons[x.downcase].to_i}
    @degeneracy = degeneracy_number
    @degeneracy
  end
35  def calc_degeneracy_w_inosine
    antal_codons = {'a' => '1', 'c' => '2', 'd' => '2', 'e' => '2',
      'f' => '2', 'g' => '1', 'h' => '2', 'i' => '3', 'k' => '2',
      'l' => '2', 'm' => '1', 'n' => '2', 'p' => '1', 'q' => '2',
      'r' => '2', 's' => '4', 't' => '1', 'v' => '1', 'w' => '1', 'y' => '2'}
40  array = @seq.split(//)
    degeneracy_number = 1
    degeneracy_number = 4 if array.last.downcase == 's'
    degeneracy_number = 2 if array.last.downcase == 'r'
    degeneracy_number = 2 if array.last.downcase == 'l'
45  array.pop
    array.each {|x| degeneracy_number *= antal_codons[x.downcase].to_i}
    @degeneracy_w_inosine = degeneracy_number
    @degeneracy_w_inosine
  end
end

```

```

def calc_frequency(n)
  freq = 100*@score.to_f/n.to_f.round.to_f/100
  freq
end
5 end
class Array
  def fetch_proteins_from(file, info)
    array = IO.readlines(file)
    var1 = 0
10   while var1 < array.length
      if array[var1].slice(0,1) == '>'
        p = Protein.new(array[var1+1])
        p.name = array[var1].chomp
        p.peptides = p.split_to_20pep if info == 20
15      p.peptides = p.split_to_15pep if info == 15
        p.peptides = p.split_to_decapep if info == 10
        p.peptides = p.split_to_nonapep if info == 9
        p.peptides = p.split_to_octapep if info == 8
        p.peptides = p.split_to_heptapep if info == 7
20      p.peptides = p.split_to_hexapep if info == 6
        p.peptides = p.split_to_pentapep if info == 5
        p.peptides = p.split_to_tetrapep if info == 4
        p.peptides = p.split_to_tripep if info == 3
        p.peptides = p.split_to_dipep if info == 2
25      self << p
      end
      var1 += 1
    end
  end
30 def find_master_prot(fetch)
    master = Protein.new("")
    self.each{|p| master = p if p.name.include?(fetch)}
    master
  end
35 end
class Time
  def Time.now_glp
    m = { "1" => "Jan", "2" => "Feb", "3" => "Mar", "4" => "Apr", "5" => "May", "6" =>
"Jun", "7" => "Jul", "8" => "Aug", "9" => "Sep", "10" => "Oct", "11" => "Nov", "12" =>
40 "Dec" }
    t = Time.now
    "#{t.day}.#{m[t.mon.to_s]}.#{t.year}"
  end
end
45 class Peptid_all_prot
  def self.calc(cut_off, limit, pep_length)
    prot_array = []
    prot_array.fetch_proteins_from("peptidcyclor_excluded_proteins2.txt", pep_length)
    array_output = []

```

```

prot_array.each do |master_prot|
  master_cut_off = master_prot.peptides.length*(1+cut_off)/limit
  selected_proteins_array = []
  peptide_array = []
5  prot_array.each do |p| #Make group of proteins similar to the master
    p.score = 0 #zero the hexa_score
    master_prot.peptides.each{|peptide| p.score += 1 if p.seq.include?(peptide)}
    if p.score > master_cut_off
      selected_proteins_array << p
10    peptide_array << p.peptides #Collect the peptides of the group
    end
  end
  peptide_array.flatten!.uniq! if peptide_array != []
  sort_peptide_array = []
15  peptide_array.each do |peptide| #Select the peptides with highest frequency
    score = 0
    prot_score = 0
    selected_proteins_array.each do |p|
      if p.seq.include?(peptide)
20      score += 1
      prot_score += p.score
    end
  end
  sort_peptide_array << [score, prot_score, peptide] if score > 1
25  end
  peptide_array = []
  if sort_peptide_array.length == 0
    peptideprofile = :nothing
  else
30    sort_peptide_array.sort!.reverse!
    sort_peptide_array.slice!(limit, sort_peptide_array.length) if
    sort_peptide_array.length > limit
    sort_peptide_array.each{|a| peptide_array << a[2]} #Best peptides in array
    peptideprofile = :evolving
35  end
  while peptideprofile == :evolving
    selected_proteins_array = []
    array = []
    prot_array.each do |p| #score the proteins against the valid peptide_array and
40  choose the best
      p.score = 0
      peptide_array.each{|peptide| p.score += 1 if p.seq.include?(peptide)}
      if p.score > cut_off #Best proteins in array
        selected_proteins_array << p
45      array << p.peptides #Their peptides in array
      end
    end
    array.flatten!.uniq! if array != []
    sort_peptide_array = []

```

```

array.each do |peptide| #find best peptides
  score = 0 #zero peptide_score
  prot_score = 0
  selected_proteins_array.each do |p|
5    if p.seq.include?(peptide)
      score += 1
      prot_score += p.score
    end
  end
10  sort_peptide_array << [score, prot_score, peptide]
  end
  peptide_array2 = []
  if sort_peptide_array.length == 0
    peptideprofile = :nothing
15  else
    sort_peptide_array.sort!.reverse!
    sort_peptide_array.slice!(limit, sort_peptide_array.length) if
sort_peptide_array.length > limit
    sort_peptide_array.each{|a| peptide_array2 << a[2]} #Best peptides in array
20  end
    if peptide_array.sort == peptide_array2.sort #If nothing happens
      peptideprofile = :finished
    else
      peptide_array = [peptide_array2].flatten
25  end
  end
  master_prot.score = 0
  peptide_array.each{|peptide| master_prot.score += 1 if
master_prot.seq.include?(peptide)}
30  array = [] #[# of members, score, name, score + number of family members, family
members, peptider]
  array << selected_proteins_array.length
  array << master_prot.score
  array << master_prot.name
35  array << "score = #{master_prot.score}, Group members =
#{selected_proteins_array.length}"
  array2 = []
  selected_proteins_array.each{|p| array2 << p.name.slice(1,5).to_i}
  array << array2
40  array << peptide_array
  array_output << array
  print "."
  end
  array_output.sort!.reverse!
45  ud_array = array_output.first[2,5]
  output_file = File.new("family_file2.txt", "w")
  array_output.each do |array|
    output_file.puts ">#{array[2].slice(1,5).to_i}"
    output_file.puts array[3]

```

```

    output_file.puts array[4].join(",")
    output_file.puts ""
  end
  output_file.close
5 puts "."
  ud_array
  end
  end
  class Peptidcycler
10 def self.calc(peptide_array, cut_off, limit)
    prot_array = []
    prot_array.fetch_proteins_from("peptidcycler_excluded_proteins2.txt", "nothing")
    excluded_file = File.new("peptidcycler_excluded_proteins2.txt", "w") #non-family
    members in back to pool
15 selected_proteins_array = []
    prot_array.each do |p| #score the proteins against the valid peptide_array and choose
    the best
      p.score = 0
      peptide_array.each{|peptide| p.score += 1 if p.seq.include?(peptide)}
20 if p.score > cut_off #de bedste proteiner i array
      selected_proteins_array << p
    else
      excluded_file.puts p.name
      excluded_file.puts p.seq
25 excluded_file.puts ""
    end
  end
  excluded_file.close
  exhausted = :no
30 exhausted = :yes if prot_array.length == selected_proteins_array.length
  [selected_proteins_array, exhausted]
  end
  end
  class Peptidegenerator
35 def self.calc(family_array, peptide_array, group_number)
    family_array.each do |p| #score the proteins against the valid peptide_array and
    choose the best
      p.score = 0
      peptide_array.each{|peptide| p.score += 1 if p.seq.include?(peptide)}
40 end
    family_array.sort!{|x,y| x.name.slice(1,5).to_i <=> y.name.slice(1,5).to_i}
    f = File.new("group_#{group_number}.txt", "w")
    family_array.each do |p|
      f.puts "#{p.name}, score = #{p.score}"
45 f.puts p.seq
      f.puts ""
    end
    f.close
    n = family_array.length

```

```

best_peptides = [] #Find best peptides
peptide_array.each do |seq|
  pep = Peptide.new(seq)
  pep.score = pep.calc_score(family_array)
5   best_peptides << pep
end
best_peptides.each do |pep| #Find properties of each hexapeptide
  position_array = []
  family_array.each{|p| position_array << p.seq.index(pep.seq)}
10  position_array.compact!
  position_array.sort!
  pep.average_position = position_array[position_array.length/2]
  pep.degeneracy = pep.calc_degeneracy
  pep.degeneracy_w_inosine = pep.calc_degeneracy_w_inosine
15  end
  best_peptides.sort!{|x,y| [y.score, x.degeneracy, x.degeneracy_w_inosine] <=>
[x.score, y.degeneracy, y.degeneracy_w_inosine]}
  output_file = File.new("group_#{group_number}_peps.txt", "w")
  output_file.puts "group #{group_number}\t#{family_array.length} proteins"
20  output_file.puts "Peptides:"
  output_file.puts "position\tsequence\tfrequency\tthits\tdegeneracy\tdegeneracy_w_l"
  best_peptides.each do |pep|
    output_file.puts
    "#{pep.average_position}\t#{pep.seq}\t#{pep.calc_frequency(family_array.length)}\t#{p
25 ep.score}\t#{pep.degeneracy}\t#{pep.degeneracy_w_inosine}"
  end
  output_file.close
  "Results for group #{group_number} written to files"
end
30 end
f = File.new("peptidcyclus_excluded_proteins2.txt", "w")
n_input_prots = 0
IO.foreach("six_families.txt") do |line|
  f.puts line.chomp
35  n_input_prots += 1 if line.slice(0,1) == '>'
end
f.close
classifier_peptides = File.new("new_family_classifying_peptides.txt", "w")
master_prot_array = []
40 score = 100
members = 100
rounds = 1
exhausted = :no
while members > 8 && rounds < 20 && exhausted == :no
45  puts "Starting round #{rounds}"
  best_family = Peptid_all_prots.calc(cut_off, limit, pep_length)
  puts "Finished peptid_all_prots10"
  fetch = "#{best_family[0].chomp}gi"
  info = best_family[1].chomp

```

```

score = info.split("=")[1].to_i
members = info.split("=").last.to_i
master_prot_array << [fetch, info]
peptide_array = best_family[3]
5 puts "Family #{rounds}"
puts "master_protein: #{fetch}"
puts "score: #{score}"
puts "members: #{members}"
array = Peptidcyclr.calc(peptide_array, cut_off, limit) #make group and peptides
10 family_array = array[0]
exhausted = array[1]
puts "Finished peptidcyclr6"
puts family_array.length
puts peptide_array.length
15 classifier_peptides.print "family #{rounds},"
classifier_peptides.puts peptide_array.join(",")
guf = Peptidegenerator.calc(family_array, peptide_array, rounds)
puts guf
rounds += 1
20 end
classifier_peptides.close
File.delete("family_file2.txt")
puts master_prot_array
output_file = File.new("parameter_variation.txt", "a")
25 output_file.puts Time.now_glp
output_file.puts "input = #{n_input_prots} sequences"
output_file.puts "limit=#{limit}\\tcut_off=#{cut_off}\\tpep_length=#{pep_length}"
output_file.puts "#{master_prot_array.length} groups"
var1 = 0
30 while var1 < master_prot_array.length
output_file.print "Group #{var1+1}\\t"
#family_array.each{|m| output_file.print "#{m.name.slice(1,5).to_i},"}
output_file.print "FAS \\t" if master_prot_array[var1][0].slice(1,5).to_i < 100
output_file.print "cycD \\t" if master_prot_array[var1][0].slice(1,5).to_i.between?(100,
35 199)
output_file.print "EDF1 \\t" if master_prot_array[var1][0].slice(1,5).to_i.between?(200,
299)
output_file.print "SP1 \\t" if master_prot_array[var1][0].slice(1,5).to_i.between?(300,
399)
40 output_file.print "PKC \\t" if master_prot_array[var1][0].slice(1,5).to_i.between?(400,
499)
output_file.print "CAT1 \\t" if master_prot_array[var1][0].slice(1,5).to_i.between?(500,
599)
#output_file.puts ""
45 output_file.puts master_prot_array[var1].last.split(",").last.strip
var1 += 1
end
output_file.puts ""
output_file.close

```

```
puts "  
output in \"parameter_variation.txt\""
```

End of “classify_family3.rb

5

The input consists of a text file (“.txt” in windows format) containing biopolymer sequences in FASTA format. In the present example the input file is called “six_families.txt” and contains 105 different protein sequences.

“classify_family3.rb” can be opened in a text editor such as notepad, SciTE, wordpad, MSword or other to define a number of parameters. The most frequently used parameters are:

cut_off: The number of selected n-mers that are present in a biopolymer should be larger than this value to include the biopolymer in the group that is defined by the n-mers.

15 limit: Number of n-mers that are selected based on frequency. E.g.; limit = 100 means that the 100 most frequently occurring n-mers will be selected.

pep_length: Length of the n-mers.

The value of these three parameters are listed in lines 1 – 3 of “classify_family3.rb”. Other parameters may be changed e.g.; file names or information written to the log file may be changed.

In the present example the parameters are:
cut_off = 9 (A protein should contain at least 10 of the selected peptides to be included in the group).
limit = 100 (The 100 most frequently occurring peptides are selected).
25 pep_length = 6 (The peptides are hexamers (six amino acids long)).

Classes, arrays, methods and other parameters and objects may be named as “amino acid”, “peptide”, “protein” or similar referring to peptide and amino acid nomenclature but the program works just as well for biopolymers and n-mers consisting of nucleotide sequences.

30 When executing “classify_family3.rb” the algorithm will generate two output files for each group of biopolymers:

One file (“group_n.txt” where n is an integer) with the selected biopolymers in FASTA format and with the score included in the name line of the sequence (The score is the number of selected n-mers that were found in the protein).

Another file ("group_n_peps.txt" where n is the same integer) with the corresponding selected n-mer sequences listed according to frequency of occurrence (frequency). In addition, the following information about each n-mer is listed:

Position: The median position in the selected biopolymers that contain the n-mer.

Hits: Number of biopolymers in the group that contain the n-mer. This is the same as frequency

Degeneracy: Number of nucleotide sequences that will encode the n-mer, last nucleotide not included if this position is degenerate,

10 Degeneracy_w_I: Same as degeneracy but with nucleotide positions that can include all four bases (A,C,G,T) substitute with an inosine that is not degenerated.

Degeneracy and degeneracy_w_I are only relevant when the n-mers are peptides.

15 "group_n_peps.txt" is a text file that can be opened as such or opened or imported into MS excel Open Office calc or another spread sheet.

In addition to the files for each group, the program will write selected information about the result to a logfile called "parameter_variation.txt".

20 In the present example, the information written to the logfile is the date of the run, number of input sequences, values for limit, cut_off and pep_length, number of groups generated, and for each group: Group number, activity of the protein used to generate the first set of hexapeptides for generation of the group and number of proteins included in the group.

Protein sequences:

25 The proteins included in the input file (accession numbers: 1705782, 224983391, 224983654, 301757408, 301785071, 296474442, 77736363, 114053227, 115496822, 260789607, 296211128, 296198149, 73997778, 54114982, 145558686, 148231179, 18858509, 113682259, 19880484, 194211581, 198443141, 2506136, 1345958, 193695213, 157115283, 301754163, 312376091, 158299938, 66515350, 30 60592790, 170587440, 309360367, 268566311, 212642053, 308499509, 73964695, 148841334, 78214939, 170038418, 292619618, 229366888, 225718944, 41152191, 225716152, 57525242, 159159985, 3043445, 225713940, 126302643, 225705832, 149546904, 225706882, 114627636, 226372738, 157817161, 209737752,

148228671, 296487921, 45709387, 45382955, 38512205, 62088936, 291389251, 114644522, 149031929, 5069468, 38511776, 148232916, 260831456, 296215214, 73963109, 292619987, 176866349, 317419553, 4885563, 28557781, 241293326, 22085162, 114653371, 291239731, 47224750, 270010958, 301757916, 209447036, 5 296203637, 73993349, 73993361, 57104988, 149730179, 224458362, 35920, 4507047, 189053509, 109120366, 4096268, 148673899, 291410388, 114649243, 114649241, 114649245, 55730091, 297693777, 18181964, 6981556 and 60302866) were found in databases provided by the National Center for Biotechnology Information (<http://www.ncbi.nlm.nih.gov>) by searching for proteins related to the 10 following protein names:

“fatty acid synthetases”, “cyclin D”, “EDF-1”, “SP1”, “PKC” and “cationic amino acid transporter 1” and sequences related to these names.

Each protein was assigned a number between “>” and “gi” in the name line of the FASTA formatted sequence. The number can be used for manual tracking of the 15 origin of the protein:

Numbers below 100: fatty acid synthetases (FAS)

Numbers between 100 and 200: cyclin D (cycD)

Numbers between 200 and 300: Endothelial Differentiation Factor 1 (EDF-1)

Numbers between 300 and 400: Sp1 transcription factor (SP1)

20 Numbers between 400 and 500: Protein Kinase C (PKC)

Numbers between 500 and 600: Cationic Amino Acid Transporter 1 (CAT1)

Results:

The input file contained animal protein sequences of six different types. 25 Between 11 and 23 proteins sequence of each type were included.

Execution of “classify_family3.rb” classified the proteins into six groups with the corresponding files:

“group_1.txt”, “group_1_peps.txt”, “group_2.txt”, “group_2_peps.txt”, “group_3.txt”, “group_3_peps.txt”, “group_4.txt”, “group_4_peps.txt”, “group_5.txt”, 30 “group_5_peps.txt”, “group_6.txt”, “group_6_peps.txt”.

The groups and the sequences were related in the following ways as shown below in table 1:

Group number	Proteins	Protein accession numbers
1	All (23) CAT1-type proteins	301757916, 209447036, 296203637, 73993349, 73993361, 57104988, 149730179, 224458362, 35920, 4507047, 189053509, 109120366, 4096268, 148673899, 291410388, 114649243, 114649241, 114649245, 55730091, 297693777, 18181964, 6981556, 60302866
2	All (20) FAS-type proteins	198443141, 2506136, 1345958, 193695213, 157115283, 301754163, 312376091, 158299938, 66515350, 60592790, 170587440, 309360367, 268566311, 212642053, 308499509, 73964695, 148841334, 78214939, 170038418, 292619618
3	All (20) cycD-type proteins	1705782, 224983391, 224983654, 301757408, 301785071, 296474442, 77736363, 114053227, 115496822, 260789607, 296211128, 296198149, 73997778, 54114982, 145558686, 148231179, 18858509, 113682259, 19880484, 194211581
4	All (17) EDF1-type proteins	229366888, 225718944, 41152191, 225716152, 57525242, 159159985, 3043445, 225713940, 126302643, 225705832, 149546904, 225706882, 114627636, 226372738, 157817161, 209737752, 148228671
5	All (14) PKC-type proteins	260831456, 296215214, 73963109, 292619987, 176866349, 317419553, 4885563, 28557781, 241293326, 22085162, 114653371, 291239731, 47224750, 270010958
6	All (11) SP1-type proteins	296487921, 45709387, 45382955, 38512205, 62088936, 291389251, 114644522, 149031929, 5069468, 38511776, 148232916

Table 1.

Conclusion.

As seen from Table 1, all sequences were classified together with the other
5 proteins of the same type as expected. This result shows that the algorithm executed
by "classify_family3.rb" and the four other scripts ("peptid_all_prots10_called.rb",
"peptidcycler6_called.rb", "peptidgenerator_all_families4_called.rb", "aminoseqLib1.rb")
is able to classify many sequences of several different types into functionally related
groups. Furthermore, the program provides a library of the most frequently occurring
10 hexapeptides for each group. This library is useful for further characterization of the
protein, for generation of degenerated primers or probes or other purposes related to
understanding the protein groups.

Example 2:

15 Classification of fungal proteins with related function and sequence in separate
groups and influence on protein classification of the parameters: number of selected
peptides (limit), number of peptides included in each group member (cut-off) and
peptide length.

Input sequences:

Names and sequences of 12 proteins with a glycosyl-hydrolase family 45 domain (accession numbers: 154294519, 222103630, 116180480, 310789959, 312217600, 39951371, 169616266, 171687659, 189197649, 27530617, 156032908 and 158138919) and 10 proteins with a glycosyl-hydrolase family 61 domain (accession numbers: 201066457, 218722209, 284451272, 2315274, 299892806, 115491813, 28919956, 119495445, 67517718 and 115391767) were used as input.

Algorithm and implementation.

10 The same program as in example 1 was used and the parameters cut_off, limit and pep_length were varied.

Any combination of the parameters that gave the result 12 gh45 proteins in group 1 and 10 gh61 proteins in group 2 were considered as a combination useful for protein classification and the peptides used for classification of each group were
15 considered as providing useful information about the group and was approved. All other results were considered as failed.

Results:

Limit	Cut_off	Pep_length	Result
100	9	6	Approved
2	0	6	Failed
3	0	6	Approved
5	0	6	Approved
6	1	6	Approved
6	1	5	Approved
6	1	4	Approved
6	1	3	Failed
7	1	6	Approved
10	2	6	Approved
10	1	6	Approved
20	3	6	Approved
50	9	6	Approved
600	7	10	Approved
1exp6	7	10	Approved
100	19	4	Approved
100	29	3	Approved
100	80	2	Approved

Table 2. Parameter testing

Conclusion.

As shown in table 2 the gh45 and gh61 proteins could be classified correctly
 5 with many combinations of the values for the three parameters tested.

The number of most frequently occurring peptides necessary to define a group
 could be as low as 3 and still lead to successful classification. For 10-mer peptides,
 only a few peptides with a frequency higher than 1 (occurring in more than 1 protein)
 were generated and the limit lost its relevancy. (Testing with a limit of 1 million
 10 peptides worked successfully).

The cut_off values that worked successfully were from 0 (one peptide
 necessary to include a protein in a group) up to cut_off = 80 peptides. In percent of the
 total number of peptides ($100 * (\text{cut_off} + 1) / \text{limit}$) the cut_off could vary from almost 0
 (7/1,000,000) to 81 %.

15 The length of the peptides that would lead to successful classification was from
 2 to 10 residues long.

Example 3:

Classification of gh61 proteins in new groups.

20 Input sequences:

Names and sequences of 467 proteins with a glycosyl-hydrolase family 61
 domain accession numbers:

74667001, 74623591, 296439555, 262118542, 296439558, 74681380, 166327,
 225557038, 150407066, 239612339, 239607981, 261199970, 261202604,
 25 291178704, 296817237, 119403851, 119403039, 119402879, 121706932,
 121699858, 121701491, 119401707, 238490450, 220691752, 238497908,
 238491658, 238503077, 220700751, 159129837, 159122044, 159123538, 66848476,
 70986426, 70994524, 66853425, 70986442, 42820662, 259486007, 67517718,
 259480946, 40740355, 40739935, 259479347, 259480639, 259487791, 75859132,
 30 40739882, 259481977, 134082518, 134080048, 145239987, 145258912, 145246562,
 145249108, 169772537, 169776393, 169772353, 83766624, 83770187, 83775441,
 83774271, 114192450, 114196513, 114192138, 114196092, 114189374, 115385899,
 115391767, 115401906, 115433194, 114192785, 115491813, 115401646,

194010899, 157679842, 154319179, 150848256, 150843601, 154291544,
154305677, 154305687, 150846167, 150843791, 154303615, 154321720,
116198863, 116197146, 88180011, 116208464, 116196852, 88179151, 116194372,
88176999, 88175803, 88177898, 116206022, 116208766, 88180297, 116201473,
5 88184980, 116178904, 88176171, 116193969, 116208324, 88178730, 88178174,
88178872, 116181126, 88182035, 116208244, 88181972, 88184289, 88179554,
88182057, 116195750, 88181172, 116202763, 116199761, 116203843, 116179468,
88178947, 116199201, 88184520, 88177588, 116196738, 116200237, 116199041,
116200816, 116203395, 119183059, 240109478, 23429037, 116503205, 116498049,
10 116497843, 299742296, 299741430, 116506365, 299744767, 169851646,
299741891, 299742644, 299753892, 169856301, 116506859, 298405205,
299747134, 116500962, 298409438, 169855583, 298408738, 298405278,
169863978, 298404712, 298405412, 169857546, 298408101, 116497409,
298405114, 298408187, 169868872, 169866035, 116504243, 298405932,
15 169856214, 299754619, 298405923, 134107111, 57223077, 46118057, 46115580,
46119467, 46139947, 46110641, 46115706, 46116252, 46124039, 46127069,
46127267, 46123465, 46123419, 46123661, 209570280, 209570302, 209570284,
209570424, 31747162, 2315274, 170102152, 170092074, 164651300, 164636998,
164642401, 170109392, 164642863, 164642075, 170105517, 170101484,
20 170105309, 145011373, 145011030, 145020107, 145014411, 145011646,
145015510, 145019304, 145016906, 145017744, 145014077, 145015931,
145021993, 39945800, 39968819, 145603548, 39946206, 149209397, 145608220,
145608962, 39972659, 145607904, 39971969, 39944092, 145605188, 145609409,
238616327, 238615335, 215456441, 238568683, 238587009, 238583365,
25 238591056, 215461462, 238579289, 238590448, 215458915, 215450835,
215451124, 238587956, 238567983, 238569868, 238579260, 215458309,
238616405, 302911456, 302883424, 302911391, 302888437, 302885549,
256726867, 302890355, 302885390, 302889367, 302887358, 256726596,
256727058, 256726169, 119498947, 119406222, 119485741, 119500958,
30 119415683, 119481757, 119495445, 119474543, 119481769, 28919725, 28920895,
85118747, 85078092, 28919956, 28924255, 28919596, 28925415, 28920933,
157071792, 85119231, 85107660, 28881165, 18376179, 16945376, 211583790,
255933578, 255945663, 255937397, 212532291, 111056092, 160705691,

160701235, 169617890, 111070506, 160706762, 111061286, 169596753,
 160703463, 169596264, 111068298, 160705400, 169616886, 111060360,
 160706840, 111065694, 169598246, 169622513, 169598063, 169594960,
 169604850, 169608836, 111069743, 160704974, 160701263, 169617193,
 5 111062780, 169619068, 160703254, 21694047, 170936818, 171693009, 171677338,
 170939885, 171685476, 170946510, 171679531, 171676648, 170946519,
 171680024, 170941538, 170936992, 170942657, 171683179, 171683736,
 170945726, 170944138, 171684255, 170941524, 170941094, 170945939,
 171694598, 171681337, 171688168, 171692645, 171690944, 171681359,
 10 171683760, 170942722, 171681569, 170942522, 170941813, 170939504,
 242218042, 242217378, 220731934, 220723726, 189198079, 187980642,
 189199012, 189194773, 189193113, 187983705, 187976621, 187983395,
 189201760, 189207084, 189188194, 189200058, 187979887, 189188372,
 187977147, 187984916, 189200631, 189192108, 189193871, 189205641,
 15 187972977, 187983372, 189191958, 187979866, 189194025, 302686954,
 300103229, 300103287, 300108602, 300101553, 300111682, 302696233,
 300105858, 302674513, 300100257, 300103639, 300106070, 302675767,
 300101263, 302683644, 302679828, 302682756, 300101194, 300103387,
 302677564, 300100552, 300100648, 302689207, 300105576, 302674561,
 20 300101299, 154700549, 156063440, 154694216, 156050139, 156045950,
 154700442, 156039846, 154700551, 156049573, 289621959, 289615175,
 289616424, 289621869, 289618672, 289622259, 289614784, 289619034,
 289615496, 289618626, 289616197, 289616196, 289617770, 289620809,
 289620945, 289620832, 289621556, 289615045, 289618715, 289618337,
 25 148553353, 218722209, 284451272, 201066457, 299892806, 302656446,
 296418037, 295636680, 237904675, 302409770, 261358989, 302420443,
 261361024, 302410193, 261352381, 261358929, 302414852, 302418676,
 261360020, 302413657, 302417124, 302405483, 261353895, 302419149,
 261358115, 302405803, 261359888, 261359952, 302405821, 261361512,
 30 302409258, 49333361, 238011426, 194704134 and 11359621
 were used as input.

Algorithm and implementation

The same program as in example 1 was used with the parameters cut_off = 9, limit = 100 and pep_length = 6.

Result:

- 5 The program was executed with the 467 gh61 proteins as input. After 13 rounds the groups became too small (five or less proteins) to define any common peptide profile because the remaining proteins had very different sequences. However, 13 round defined the groups as listed in table 3. Each group had its own profile of hexapeptides (100 most frequently occurring hexapeptides) with little overlap between
- 10 groups as illustrated in table 4.

Group	Proteins
1	86
2	43
3	27
4	24
5	19
6	18
7	15
8	15
9	14
10	14
11	13
12	10
13	9

Table 3. Groups of gh61 proteins

Subfamily	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100	3	3	1	0	0	0	5	1	2	3	0	2
2	3	100	12	0	0	1	1	7	0	5	2	1	4
3	3	12	100	7	0	2	0	4	2	5	3	0	3
4	1	0	7	100	0	1	0	0	2	1	0	1	1
5	0	0	0	0	100	0	0	0	0	0	0	0	0
6	0	1	2	1	0	100	0	1	6	2	0	0	1
7	0	1	0	0	0	0	100	0	0	0	3	1	0
8	5	7	4	0	0	1	0	100	0	3	6	1	2
9	1	0	2	2	0	6	0	0	100	0	0	0	0
10	2	5	5	1	0	2	0	3	0	100	2	0	1
11	3	2	3	0	0	0	3	6	0	2	100	0	1
12	0	1	0	1	0	0	1	1	0	0	0	100	0
13	2	4	3	1	0	1	0	2	0	1	1	0	100

- 15 Table 4: Cross comparison of the hexapeptide signatures for each group (group) of gh61 proteins.

Conclusion.

- The distribution of the hexapeptides showed that almost all the conserved
- 20 peptides were found in the first 240 amino acids of the gh61 proteins whereas the

remaining part of the proteins was highly variable. Interestingly, two peaks were observed where most of the conserved hexapeptides were found for all the family. Region 1 located between amino acid residues 100-120 has a clear peak whereas region 2 (amino acids 160-200) has a shoulder at amino acids 200-240.

5 The distribution of peptides for each group shows that groups 1, 5, 7, 8, 9, 10 and 12 (type 1) only have peaks in regions 1 and 2 whereas groups 2, 3, 4, 6, 11 and 13 (type 2) have an additional peak from amino acids 200-220. Peak 1 was very small in group 7, indicating that this group is poorly conserved in region 1.

 To investigate the three conserved regions in more detail we aligned the
10 hexapeptides for each group in these regions to generate a consensus sequence. Alignment of the consensus sequences for each group shows that region 1 is similar in 11 of the 13 groups and contains one of the histidines that coordinates the divalent nickel atom bound to the gh61 crystal structure (Karkehabadi, S. et al., 2008. The first structure of a glycoside hydrolase family 61 member, Cel61B from *Hypocrea jecorina*,
15 at 1.6 Å resolution. *Journal of Molecular Biology*, 383(1), 144-154). Furthermore, the two cysteines that form a cysteine bridge in the crystal structure were found in this region but were only conserved in groups 1, 8 and 11. In group 11, the second of the two cysteines was found in the protein sequences but was located outside the conserved hexapeptides.

20 Region 2 was conserved in all 13 groups and contains a conserved histidine that does not participate in coordination of the nickel atom (Karkehabadi, S. et al., 2008. The first structure of a glycoside hydrolase family 61 member, Cel61B from *Hypocrea jecorina*, at 1.6 Å resolution. *Journal of Molecular Biology*, 383(1), 144-154) but nevertheless is located on the nickel-binding surface of gh61 together with two
25 other conserved residues (Q/E₄₉ and Y₅₁) in region 2. Region 3 is outside the reported crystal structure and contains a conserved proline-glycine dipeptide.

 To further compare the groups we aligned the proteins that gave the highest score in each group. Notably, this alignment showed that His₂₃ that participates in binding to the nickel atom was conserved in all families. Also the two cysteines (Cys₇₈
30 and Cys₂₂₈) that form a cystein bridge in the crystal were conserved. Interestingly, the alignment is poor between amino acids 60 and 80 where several residues map to the nickel binding surface of gh61.

Thus the herein described algorithm programmed as in example 1 is able to divide a large number of highly divergent gh61 family proteins into comprehensible groups.

Example 4:

- 5 Generation of degenerated primers for PCR of plant Absciscic Acid Responsive Element (ABRE) binding factors.

Input sequences:

A library of 92 amino acid sequences that were characterized as ABRE binding factors (ABF) or had high sequence similarity to ABFs (accession numbers

10 215272920, 297820460, 297818916, 297827105, 297847296, 10120429, 18404091, 13346153, 18407786, 15230146, 18405590, 7635468, 13346155, 30693958, 92090805, 92090809, 92090803, 92090807, 92090811, 168988212, 167882612, 2228771, 3273764, 312205693, 111115690, 27469352, 111115692, 302398623, 59896064, 87162867, 14571808, 67906424, 125554512, 218202273, 125541258,

15 218197011, 218189417, 115479537, 55297501, 222635185, 125605945, 115464587, 222619435, 297597837, 115448891, 20161640, 124055245, 224102821, 118483091, 224100543, 224113323, 224087391, 224113321, 224107267, 302746499, 255575677, 255565609, 255542297, 255538288, 302771984, 302763137, 302817392, 302790590, 302774659, 302795271, 42561991, 242055043, 242044916,

20 242059039, 172052530, 62898533, 167745176, 172052528, 117307414, 172052532, 21693585, 306450635, 146275401, 147819811, 296084920, 225463745, 225440904, 147805356, 225464964, 297740105, 195659033, 226507468, 308081734, 194701146, 212723434, 308044319, 195652633)

- 25 Algorithm and implementation

The same program as in example 1 was used with the parameters cut_off = 9, limit = 100 and pep_length = 6.

Design of degenerated primers

- 30 Conserved hexapeptides identified were reverse translated according to the genetic code and positions containing any nucleotide (A, C, G or T) were substituted with inosine (Table 5). Degenerate nucleotides at the 3' end of the primers were removed from the sequence of the primers. The degeneracy of the primer that results

from reverse translation of each hexapeptide was calculated based on the genetic code and substituting positions containing any nucleotide (A, C, G or T) with inosine (Table 5). In addition, the relative position of the hexapeptides in the proteins was estimated as the median of the distance of the peptide to the N-terminal of each protein in the subgroup that contained the peptide.

Sequences for primers were selected on three criteria:

1. They should have high frequency in the subfamily of proteins
2. They should give an amplicon of at least 40 base pairs excluding primer sequences in order to be able to get sufficient sequence information to identify the PCR product.
3. The primers should have the smallest possible redundancy and redundant bases at the 3' end are not allowed.

A tail of six bases (CTGGAC) was added to the 5' end of all primer sequences as this is reported to improve the performance of short primers.

Reverse primers were designed to be reverse complementary to the DNA sequence encoding the hexapeptide and according to the same rules.

In silico PCR

Performed by searching in protein sequences for hexapeptide sequences defined as forward or reverse primers. A protein sequence is defined as found if it contains a sequence corresponding to at least one of the forward primers and a sequence corresponding to one of the reverse primers.

Results:

All of the input sequences were classified into one group with a common set of hexapeptides that could be used for generation of primers.

Four hexamers (table 5) were chosen for generation of primers based on the chosen criteria such as high frequency, low degeneracy (with inosine incorporation) and generation of an amplicon of an informative length.

Sequence	Position	Frequency	Degeneracy _w_l	DNA sequence	Primer type
RESAAR	282	92	32	ctggacckigcigciswytic	Reverse
RKQAYT	291	80	16	ctggacgtrtaigcytyttic	Reverse
GENTLE	137	73	4	ctggacggigaratgacictiga	Forward
TVDEVW	106	65	4	ctggacacigtigaygargtitgg	Forward

Table 5. Hexamer peptides chosen for generation of degenerated primers

Position: The median position of the hexapeptide in the proteins that contain the hexapeptide.

Frequency: Number of proteins that contain the hexapeptide.

- 5 Degeneracy_w_l: Number of nucleotide sequences that will encode the n-mer, last nucleotide not included if this position is degenerate but with nucleotide positions that can include all four bases (A,C,G,T) substitute with an inosine that is not degenerated. DNA sequence: Sequence of the degenerated primer design according to the design criteria.

- 10 Primer type: Designates whether the primer will be used as forward or reverse primer in PCR.

In silico PCR with RESAAR and RKGAYT as reverse primers and GEMTLE and TVDEVW as forward primers found 87 of the 92 input ABF sequences (94 %).

This example demonstrates that the algorithm can be used to find peptide sequences

- 15 suitable for degenerate primers of a family of plant transcription factors without aligning the input sequences.

Example 5:

Find peptides suitable for generation of degenerated primers for a DNA polymerase for

- 20 PCR detection of *Streptococcus pneumonia*.

Input sequences:

- Names and sequences of 31 DNA polymerases from *Streptococcus pneumonia* (accession numbers: 24376673, 307126259, 225857894, 306408250, 220673442, 25 303254143, 303267731, 237822599, 183573486, 183576083, 182683002, 116516654, 194357058, 169833568, 301793347, 301799217, 225725792, 302637669, 149004186, 149004189, 147761726, 149010899, 183572034, 147929795, 149023475, 148985377, 148987754, 147926898, 14971495, 153765, 13621480) and 2 putative exonucleases from *Salmonella enterica* (accession number: 30 16505991) and *Yersinia pestis* (accession number: 5834736).

Algorithm and implementation

The same program as in example 1 was used with the parameters cut_off = 9, limit = 100 and pep_length = 6.

Design of degenerated primers

- 5 Primers were designed with the same criteria as in example 4.

In silico PCR

Performed as described in example 4.

10 Results:

- 25 of the input sequences from Streptococcus pneumonia were classified into one group with a common set of hexapeptides that could be used for generation of primers. Neither the protein from Salmonella enterica not the protein from Yersinia pestis were included in the group and the remaining six sequences from Streptococcus pneumonia no significant sequence similarity to the the 25 sequences in group 1.

Two hexamers (table 6) were chosen for generation of primers based on the chosen criteria such as high frequency, low degeneracy (with inosine incorporation) and generation of an amplicon of an informative length.

20

Sequence	Position	Frequency	Degeneracy _w_l	DNA sequence	Primer type
YEADDI	112	25	16	ctggactaygargcigaygayat	Forward
MKENLI	230	25	16	ctggacatiagrtytycttcat	Reverse

Table 6. Hexamer peptides chosen for generation of degenerated primers

Position: The median position of the hexapeptide in the proteins that contain the hexapeptide.

- 25 Frequency: Number of proteins that contain the hexapeptide.

Degeneracy_w_l: Number of nucleotide sequences that will encode the n-mer, last nucleotide not included if this position is degenerate but with nucleotide positions that can include all four bases (A,C,G,T) substitute with an inosine that is not degenerated.

- 30 DNA sequence: Sequence of the degenerated primer design according to the design criteria.

Primer type: Designates whether the primer will be used as forward or reverse primer in PCR.

In silico PCR with YEADDI forward primer and MKENLI as reverse primer found the 25 sequences in group 1.

- 5 This example demonstrates that the algorithm can be used to find peptide sequences suitable for degenerate primers for bacterial proteins that can be used for detecting the bacterium.

Example 6

- 10 Find nucleotide sequences suitable for generation of primers for detection of mammalian reovirus.

Input sequences:

- Names and sequences of 206 mammalian reovirus nucleotide sequences
- 15 (accession numbers: 40352701, 23309024, 14211779, 14211783, 14211777, 14211775, 117574672, 14211771, 23307872, 23307895, 300428967, 5353960, 23307868, 116794829, 23307891, 300428968, 333717, 258549717, 257220490, 294854040, 499863, 32479528, 116266546, 23307906, 300428966, 258549709, 21708063, 42542874, 114107200, 23307874, 61920, 333679, 40886743, 19851957,
- 20 40886741, 40886739, 40886733, 40886735, 40886745, 45268957, 40886747, 40886749, 40886737, 28175371, 14149149, 300428969, 258549713, 6467504, 257220498, 6467508, 90018315, 6467506, 333689, 300428971, 23309030, 333638, 333636, 333659, 94435331, 23307876, 785075, 703264, 300428970, 258549715, 333652, 23307882, 808811, 808813, 333671, 83656962, 155199109, 155199115,
- 25 155199111, 155199117, 77386241, 300428962, 61780, 57903312, 115112426, 333742, 57903314, 57903316, 57903310, 31790195, 57903320, 57903318, 257220500, 4887001, 310289, 23307885, 77386247, 90018311, 310287, 23309032, 310285, 258549723, 300428963, 333747, 3170622, 3170624, 132424637, 132424641, 117574671, 3170632, 257220504, 23309040, 23307880, 333755,
- 30 77386243, 333757, 90018317, 62911145, 12751410, 90018313, 12751412, 23307908, 333759, 62911143, 61947, 62911141, 333719, 62911137, 62911135, 155037964, 77386245, 3170638, 3170630, 300428964, 300428965, 3170640, 45433513, 9931305, 257220506, 132424639, 132424643, 115279670, 11323275,

11323271, 11323281, 11323273, 18031637, 18031611, 18031609, 18031607,
 18031605, 18031597, 18031589, 18031587, 18031585, 18031583, 18031581,
 18031579, 18031573, 18031571, 18031569, 18031567, 18031639, 18031563,
 18031615, 18031617, 18031619, 18031625, 18031627, 18031629, 18031641,
 5 18031565, 14669464, 14669456, 14669462, 14669458, 14669460, 73533672,
 73533662, 73533664, 73533674, 73533668, 73533676, 73533658, 73533678,
 163639398, 333705, 333693, 333647, 333666, 333713, 333761, 333634, 333716,
 333658, 333661, 333700, 333754, 333683, 333663, 333688, 333670, 333763,
 333711, 333642, 333651, 333668, 333686, 333657, 333643, 333633, 333678).

10

Algorithm and implementation

The same program as in example 1 was used with the parameters cut_off = 10, limit = 100 and pep_length = 20.

15 *In silico* PCR

Performed by searching in nucleotide sequences for shorter nucleotide sequences defined as forward and reverse primers. A nucleotide sequence is defined as found if it contains a sequence corresponding to at least one of the forward primers and a sequence corresponding to one of the reverse primers.

20

Results:

Reovirus contains 10 – 12 DNA segments that can be considered as independent viral chromosomes. Thus any structurally meaningful conserved sequence is expected to map to a specific segment whereas conserved sequences
 25 found on different segments are expected to occur by chance.

24 of the input sequences from reovirus were classified into the largest group. These sequences (accession numbers: 294854040, 499863, 32479528, 258549709, 18031637, 18031611, 18031609, 18031607, 18031605, 18031597, 18031589, 18031587, 18031581, 18031579, 18031639, 18031563, 18031615, 18031617,
 30 18031619, 18031625, 18031627, 18031629, 18031641, 18031565) contain sequences from the Lambda-3 gene on the L1 segment of reovirus and are thus from the same segment as expected. Likewise, group 2 (accession numbers: 114107200, 23307874, 61920, 333679, 40886743, 19851957, 40886741, 40886739, 40886733, 40886735,

40886745, 45268957, 40886747, 40886749, 40886737, 300428969, 258549713, 333634, 333658) are sigma-3 gene sequences from segment S4.

Four primers were chosen (table 7) based on generation of an amplicon of an informative length. These primers found 21 of the 24 nucleotide sequences in group 1 (87 %).

Sequence	Position	Frequency	Primer type
GCCGATATCGGGAATGCAGA	22	22	Forward
ATGCAGAACATGATTCAGCA	35	20	Forward
TCTACTGAGCATACTGCTAA	167	19	Reverse
TCTTTTCTCCAGGTAACGA	101	18	Reverse

Table 7: 20-mer nucleotides chosen as primers

10 Position: The median position of the 20-mer nucleotide in the DNA sequence that contains the 20-mer.

Frequency: sequence that contains the 20-mer.

Primer type: Designates whether the primer will be used as forward or reverse primer in PCR.

15 Thus this method can classify reovirus segments into structurally meaningful groups and makes it possible to find conserved nucleotide sequences that can be used directly or their reverse complementary sequence as primers for amplification of different reovirus segments.

20 Example 7:

Generation of primers to isolate new gh61 genes belonging to group 1.

Design of degenerated primers

Conserved hexapeptides identified in example 3 were reverse translated and
25 primers were designed with the same criteria as in example 4

The primers were synthesized and HPLC-purified by Sigma-Aldrich (UK/Europe).

Peptide	Primer name	Final primer sequence
DIICHK	61.1	CTGGACGAYATHATHTGYCAYAA
EIIALH	61.2	CTGGACTGIAGIGCDATDATYTC
HHGPV	61.3	CTGGACCAYCAYGGICCI GT
GAQNYP	61.4	CTGGACGGRTARTTYTGIGCICC
LEFFKI	61.5	CTGGACCTIGARTTYTTYAARAT

Table 8: List of degenerated primers.

DNA purification

5 DNA was isolated from 14 thermophilic fungi of different families obtained from The Centraalbureau voor Schimmelcultures, The Netherlands and grown on 6% wheat bran (Finax, Denmark), 15% agar (Sigma-Aldrich, UK/Europe) plates at the recommended temperature.

Fungal mycelium was first scraped of the top of a wheat bran agar plate, frozen
10 in N₂(l) and grinded with a mortar and pestle. DNA was extracted from the homogenized mycelium with the Fungal DNA Mini Kit (Omega Bio-Tek, USA) according to the manufacturer's instructions.

PCR was performed using standard conditions and the PCR products were cycle sequenced by Eurofins-MWG (Germany) or StarSEQ (Germany) with one of the
15 degenerated primers used for PCR. The resulting sequences were translated to amino acid sequence and used for BLAST search against the non-redundant protein sequence database at NCBI and inspected for conserved domains (Marchler-Bauer et al., 2009) in the CDD database at NCBI to identify sequences encoding glycohydrolase family 61-like proteins.

20

Results

The most frequently occurring hexapeptides defining group 1 of gh61s were used for design of degenerated primers (Table 8). As the two most conserved hexapeptides (occurring in 80 and 78 % of the proteins) could be used for design of
25 reverse primers we did not find it necessary to design a third reverse primer. One of the three hexapeptides used for forward primer design (SHHGPV) contains one serine residue that is coded by 6 different codons at the N-terminal. A degenerate primer to serine does not contribute significantly to specificity and therefore, the primer was made by reverse translation of the peptide HHGPV. In *in silico* PCR the three forward

and two reverse primers were able to amplify 66 of the 85 proteins in group 1 and no proteins from other groups.

The primers were used for all six possible combination for PCR of DNA from the 14 thermophilic fungi. For all the fungi at least one of the primer sets gave an amplification product with the expected size and for some fungi all the primer sets gave a positive product. For each fungus, the longest amplicon that had the expected size was sequenced and analyzed for open reading frames. All the amplicons yielded a sequence that encodes a novel, putative GH61 gene fragment. Although the isolated sequences are only partial, it was possible to classify all except one as belonging to group 1. The unassigned sequence from *C. senegalense* was the shortest of the sequences and is only 37 amino acids long but had up to 73 % identical to known gh61 sequences and 78 % to the new sequence from *R. thermophila*. In summary, the PCR result showed that degenerated primers based on the hexapeptide finder algorithm could be used to find new gh61 proteins.

Example 8:

Alignment of new sequences from unordered, short sequences and extraction of a sequence related to a known protein family.

Input sequences:

A library of 51 amino acid sequences that were annotated as glycosyl-hydrolase family 45 proteins in CAZY (accession numbers 189577959, 62770092, 62770095, 62770085, 520823, 62770091, 6179891, 151303713, 151303715, 38492164, 26516781, 40739414, 222103626, 222103630, 224434578, 224434580, 37732125, 126697302, 4249556, 4249558, 4756863, 39951371, 32526553, 62821724, 62821722, 15384734, 28881412, 194143489, 4210808, 195547039, 56410394, 238033880, 8052314, 170943791, 170936906, 8926975, 27530542, 27530617, 27530615, 116001534, 197267671, 158138919, 86476841, 116001536, 49731514, 33521684, 1478379, 4756871, 4756879, 4756877, 4756875) but not the sequence from *Melanocarpus albomyces* (accession number 27125829).

A library with a nucleotide sequence from *Melanocarpus albomyces* annotated as glycosyl-hydrolase family 45 protein in CAZY (accession number 27125829) and a

nucleotide sequence from *Melanocarpus albomyces* annotated as glycosyl-hydrolase family 7 protein in CAZY (accession number 27125836).

Algorithms

5 Generation of conserved hexapeptides:

1. For each GH45 input sequence make all the hexamers that occur in the biopolymer sequence.
2. For each hexapeptide: Count the number of GH45 input sequences that has at least one occurrence of the hexapeptide in its sequence and assign this number to the hexapeptide. This is the hexapeptide frequency.
- 10 3. Select the 18 hexapeptides with the highest frequency. If two hexapeptides have the same frequency and only one can be included, pick one at random.

15 Generation of random 34-mers:

A library of 34-mers starting at random positions in the input sequences was generated. The coverage was ten times defined as that the total number of nucleotides in the library of random 34-mers was ten times the number of nucleotides in the library of input sequences.

20

Selection of random 34-mers:

All the random hexamers and their complementary sequence were translated in all three reading frames into peptides. Any peptide with a stop codon was discarded.

- All 34-mer encoding a peptide in any of the three reading frames containing one of the
- 25 conserved hexapeptides were selected.

Assembly of selected 34-mers:

- All selected 34-mers were assembled into longer sequences in all possible combinations if they overlapped by at least 17 nucleotides. These longer sequences
- 30 were further extended if they overlapped by at least 17 nucleotides with each other or with a 34-mer. E.g.; if the letters A, B, C and D represent sequences that are 17 nucleotides long, then a first 34-mer with the sequence AB overlaps with a 34-mer with the sequence BC but not with a 34-mer with the sequence CB. AB and BC will form the

contig ABC, which overlaps with CD but not with AC. Thus ABC and CD can be assembled to the contig ABCD. Likewise, the contigs ABC and BCD can be overlapped to form ABCD. This process was continued until no new nucleotide sequences were generated.

5

Extension of assembled 34-mers

The assembled 34-mers were used as template and extended by 34-mers from the pool of random 34-mers in all possible combinations if they overlapped by at least 17 nucleotides. These longer sequences were further extended if they overlapped by at least 17 nucleotides with each other or with a 34-mer from the pool of random 34-mers E.g.; if the letters A, B, C, D and E represent sequences that are 17 nucleotides long, then the contig ABCD can be overlapped with a 34-mer with the sequence DE from the pool of random 34-mers to form the contig ABCDE. This process was continued until no new nucleotide sequences were generated.

15

Results

A library of the 18 most frequently occurring hexapeptides in the group of 51 glycosyl-hydrolase family 45 proteins from CAZY was generated. These are the conserved hexapeptides.

20

The library with the two nucleotide sequences from *Melanocarpus albomyces* was used to generate a library of random 34-mer sequences. This library can be viewed as a file containing 34 nucleotides sequencing reads with a ten times coverage of the two input sequences and where the gh7 and the gh45 sequences are mixed at random.

25

To find the sequences encoding a gh45 in the library of random sequences all the random 34-mer that encoded any of the conserved hexapeptides were selected, assembled and further extended with sequences from the pool of random 34-mers.

The result of the assembly contained three sequences:

The longest was 405 nucleotides and identical to nucleotides 194 – 599 of the GH45 from *M. albomyces* (accession number 27125829).

30

The second sequence was 263 nucleotides and identical to nucleotides 194 – 462 of the GH45 from *M. albomyces* (accession number 27125829) except for a gap at nucleotides 419- 423.

The last sequence was 126 nucleotides and identical to nucleotides 1 – 126 of the GH45 from *M. albomyces* (accession number 27125829).

None of the sequences had any similarity to the GH7 from *M. albomyces* (accession number 27125836).

5 The best hits (that are not from *M. albomyces*) of a BLAST search with the 405 and the 126 nucleotides long sequences are sequences (accession numbers: 171677675 and 4249557) that encode proteins with GH45 domains. Moreover, when the assembled nucleotide sequences are translated into protein they are themselves recognized as GH45 domains by conserved domain search (Marchler-Bauer A et al.
10 (2011), Nucleic Acids Res.39(D)225-9).

In conclusion the assembly of artificially generated, random sequence reads based on conserved hexapeptide sequences from GH45s yielded extended nucleotide sequences that can easily be identified as gh45-related sequences.

These sequences can be used for further extension *in silico* by other methods
15 or as probes, primers or for other ways of further studies of the sequences and the putative proteins they encode.

Example 9

Classification of functionally characterized GH5 proteins and assignment of function-
20 related subfamilies.

Input sequences:

118 eukaryote proteins classified as characterized GH5 proteins in the Carbohydrate Active Enzymes database (www.cazy.org) (Cantarel et al. 2009) (accession numbers: 14119788, 14119613, 167276395, 60300013, 2997731, 3242653, 15054480,
25 15054476, 12247839, 55847696, 197313035, 257180, 164414992, 95025917, 95025694, 4586414, 59932919, 2654525, 2624931, 22652206, 22652210, 296803329, 3513514, 3098605, 15822650, 3098607, 1304102, 505195, 170549, 55295400, 677887, 1929906, 20543838, 13398416, 13398418, 5639669, 3712668, 2353007, 2148986, 2353005, 2935581, 1688087, 170083838, 163644902,
30 239580117, 984166, 224459236, 8052316, 5457159, 19773501, 6855474, 60360860, 17017348, 16356671, 38175312, 39726088, 4062993, 187766738, 12584219, 189498328, 53955310, 1064880, 83767960, 20385851, 7638024, 95025945,

1150442, 62945154, 19920182, 1150406, 238031158, 51830530, 171491, 861117,
 1150615, 4007667, 228204855, 1150694, 51173868, 95025907, 116766026,
 190663747, 19072999, 116766034, 22655548, 3687495, 5531467, 115834714,
 59766689, 1679597, 82659769, 558311, 66845127, 197293759, 197293757,
 5 134075158, 210077419, 84621433, 197260976, 10178872, 157703990, 95025929,
 95025825, 95025864, 85658727, 506848, 256002459, 29786430, 9836826,
 18413497, 15529298, 110627661, 294653278, 308097160, 37544517, 21307619,
 54299552, 307776646)

Algorithm and implementation

10 The same program as in example 1 was used and the parameters cut_off, limit and pep_length were varied.

Furthermore, a peptide cut off of > 0.2 was used. The frequency of each peptide is defined as the number of proteins (from the group of proteins) that include the peptide (e.g., 21 out of 100 proteins; frequency = 0.21). Each peptide should have a frequency
 15 higher than 0.20 to be included on the peptide list otherwise it is discarded. Thus, if the limit (maximum number of peptides included on the list) is 70 only 69 peptides will be included if the frequency of peptide number 70 is 0.20 or less.

Moreover, all peptides with the same frequency as the peptide ranked as #71 were removed. For example, if the frequency of the peptide ranked as #71 was 0.35 and
 20 peptides #69 and 70 also had the frequency 0.35, these two peptides were removed and the final peptide list consisted of only 68 peptides.

For each round of PPR, the largest group of proteins was removed as a subfamily and the analysis was repeated with the rest of the proteins.

Each protein subfamilies was assigned a function corresponding to the function of
 25 the most abundant enzyme type in the subfamily.

Assignment of the 118 GH5 proteins to subfamilies and function.

Each of the 118 GH5 proteins were given a score for each subfamily-specific peptide lists by:

1. Finding all the peptides from the list that were present in the sequence of the
 30 protein.

2. Sum the frequency of these peptides. This gave the subfamily-specific frequency score.

The proteins were assigned to the subfamily with the highest subfamily-specific frequency score and the function assigned to subfamily was taken as the function

5 predicted for the protein.

Results:

The 118 eukaryotic GH5 proteins have very divergent sequences with an average pair wise identity of 9 % and only 23 % of the pair wise sequence comparisons producing any significant alignment. When these proteins were analyzed by PPR with

10 different parameters (peptide length, number of peptides and cut off) each analysis resulted in a number of protein subfamilies that were assigned a function corresponding to the function of the most abundant enzyme type in the subfamily. E.g. a subfamily with 16 glucan 1,3- β -glucosidases and one glucan endo-1,6- β -glucosidase was assigned the function glucan 1,3- β -glucosidases ignoring the single enzyme with

15 another function. Next, the all of the 118 GH5 proteins were scored against the subfamily-specific peptide lists and placed in the subfamily with the highest score. Finally, the function of the proteins as reported in CAZy was compared to the function assigned to the subfamily where the protein was placed. The parameters tested were from peptides of length 3 -10, peptide lists with 30 -200 conserved peptides and cut off

20 from 5 – 40 peptides. The cut off is the number of peptides from a list of conserved peptides that a protein should contain to be part of the subfamily. The cut off and the number of peptides on the list are related in the way that a higher cut off for the same peptide list will lead to inclusion of fewer proteins, whereas a longer peptide list for the same cut off will include more proteins. It is convenient to describe this relationship as

25 a stringency parameter and define stringency as cut off divided by number of peptides. For example, cut off = 5 and number of peptides = 50 gives a stringency of 0.10 whereas cut off = 20 and number of peptides = 100 gives a stringency of 0.20.

For the 118 GH5 proteins the number of enzymes that were assigned to subfamilies with the same function (% correct classification) depended on the stringency of the

30 PPR analysis (figure 1A) with a very broad maximum: Stringencies between 0.10 and 0.33 correctly classified 78 – 86 % of the 118 proteins when averaging the results for

the peptide lengths from 3 – 10 mers. However, peptides of length 4- 6 amino acids were better for classification of the GH5 proteins according to function than shorter or longer peptides (figure 1B). For 4-6 mer peptides there was also a relative broad range of optimal stringency that gave correct classification of most of the enzymes. The
5 highest level of correct classification was observed with hexamer peptides at a stringency of 0.14 (figure 1C). To test the influence of the number of conserved peptides on the classification of the GH5 proteins we tested this parameter with hexamer peptides and a fixed stringency of 0.14. The result showed that the highest level of correct classification was obtained with 70 – 100 peptides but also other
10 numbers of peptides gave a high correct classification of more than 80 % of the enzymes (figure 1D).

The PPR parameters that gave the best classification of the GH5 proteins into functional subfamilies was hexamer peptides, number of peptides = 70 and cut off = 10, (stringency = 0.14). PPR analysis with these parameters generated nine
15 subfamilies containing 97 of the GH5 proteins (figure 2A). Both the cellulases (EC 3.2.1.4) and the mannan endo- β -1,4-mannosidase (EC 3.2.1.78) were divided into several groups whereas the glucan 1,3- β -glucosidase (EC 3.2.1.58), and glucan endo-1,6- β -glucosidase (EC 3.2.1.75) were placed in a single group for each enzyme type. One glucan endo-1,6- β -glucosidase (EC 3.2.1.75) was placed together with 16 glucan
20 1,3- β -glucosidase (EC 3.2.1.58). Except for this proteins, PPR generated subfamilies that correlated with the function of the GH5 proteins. Cross comparison of the hexapeptide lists for the nine groups showed that very few of the peptides were shared between subfamilies (figure 2B). The largest overlap between subfamilies was of only three peptides between subfamilies 2 and 8.

25 When the 118 GH5 proteins were scored against these subfamily-specific peptide lists 115 of the proteins (97 %) were assigned to a subfamily containing proteins with the same function (subfamily 2 function defined as EC3.2.1.58), (figure 2C).

PPR classified a GH5 endocellulase (EC3.2.1.4) from *Thermoascus aurantiacus* (accession number 16356671) to subfamily 1 (figure 2D). Eleven functionally important
30 amino acid residues can be inferred from the crystal structure of this enzyme (Van Petegem et al. 2002; Lo Leggio & Larsen 2002). Interestingly, ten of these functionally

important amino acids including the two catalytic glutamates, were found in the 70 most frequent hexapeptides for subfamily 1 (Peptide list with sequence1, frequency; sequence2, frequency...:

QYLDSD,0.95;MHQYLD,0.95;HQYLD,0.9;LNQAAI,0.86;WAAGPW,0.76;SDGSGT,0.
 5 76;WWAAGP,0.76;AAGPWW,0.76;LDSDGS,0.76;DSDGSG,0.76;YLDSDG,0.76;AGP
 WWG,0.71;FDTNNE,0.71;EMHQYL,0.71;ATSQYI,0.67;GATSQY,0.67;DGSPTS,0.67;
 TGAWTW,0.67;IRAAGA,0.67;RAAGAT,0.62;AGATSQ,0.62;NLNQAA,0.62;YIFVEG,0.
 57;AAGATS,0.57;QYIFVE,0.57;IFVEGN,0.57;YEMHQY,0.57;PHNYGR,0.57;GEFAGG
 ,0.57;DTNNEY,0.57;DPHNYG,0.52;WTGAWT,0.52;SGAEFG,0.52;GPWWGD,0.52;S
 10 WTGAW,0.52;TSQYIF,0.52;PWWGDY,0.52;ESGAEF,0.52;SQYIFV,0.48;IFDTNN,0.48
 ;VIFDTN,0.48;EGNSWT,0.48;NSWTGA,0.48;GNSWTG,0.48;FVEGNS,0.48;HNYGRY,
 0.43;GAWTWT,0.43;VYEMHQ,0.43;GALWWA,0.43;NESGAE,0.43;LNLNQA,0.43;GIR
 AAG,0.43;NQAAIN,0.43;NQAAID,0.43;VLNLNQ,0.43;TNNEYH,0.43;LVLNLN,0.43;LW
 WAAG,0.43;VEGNSW,0.43;NEYHDM,0.38;LGEFAG,0.38;GMNIFR,0.38;EFAGGA,0.3
 15 8;NNEYHD,0.38;IVYEMH,0.38;KIVYEM,0.38;FAGGAN,0.38;) suggesting that also the
 residues surrounding the functional amino acids are somewhat conserved. However,
 one of the hexapeptides containing a functionally important amino acid was only found
 in 6 of the 21 proteins and none of the hexapeptides were found in all of the proteins.

20 Conclusion:

The results of the subfamily classification of the functionally characterized GH5 proteins suggest that PPR subfamilies can be used to predict the function of complicated protein families such as GH5 with high accuracy (in this case up to 97 %).

25 Example 10.

Generation of function-specific peptide lists from functionally characterized GH5 proteins and use of the lists prediction of the function of other gh5 proteins.

Generation of function-specific peptide lists from GH5 proteins

30 The same 118 eukaryotic GH5 proteins as in example 9 were divided into four lists of proteins with the same function: cellulase (EC 3.2.1.4), glucan 1,3- β -glucosidase (EC 3.2.1.58), glucan endo-1,6- β -glucosidase (EC 3.2.1.75) and mannan endo- β -1,4-

mannosidase (EC 3.2.1.78). Half of the protein sequences chosen at random from each list were used for step 3 of the PPR algorithm: Make all the n-mer peptides that occur in these proteins and find the N most frequently occurring peptides where N is a predefined number of n-mer peptides. In this way, peptide patterns were created for
5 the four enzyme classes.

The other half of the GH5 proteins were assigned a function by:

1. Finding all the conserved peptides for each subfamily that were present in the sequence of the protein.
2. Sum the frequency of these peptides. This gave the subfamily-specific
10 frequency score.
3. Assign the protein to the subfamily with the highest subfamily-specific frequency score.

This was done ten times by randomly choosing half of the GH5 proteins belonging to each of the enzyme classes EC 3.2.1.4, 3.2.1.58, 3.2.1.75 and 3.2.1.78. Each time
15 the peptide lists was used to predict the function of the other half of the proteins. Different parameters (n-mer length and number of peptides, N) were tested.

Results:

The results obtained by this implementation of PPR show that it was possible to correctly predict the function of 92 % of the proteins with 98 % accuracy (figure 3A-C).
20 Interestingly, the highest, correct prediction rates were achieved with a large number (400-3200) 3-5 mer peptides (figure 3C). When the analysis was performed with all possible peptides of a given length it was also possible to correctly predict the function of 93 % of the enzymes with 3-5 mer peptides (figure 3D).

Conclusion:

25 A library of proteins with the same functions can be used to make a library of n-mer peptides and calculate their frequency. This library of n-mers can predict the function of complicated protein families such as GH5 with high accuracy (more than 90 %).

Example 11. Peptide Pattern Analysis of many proteins from the GH13 CAZy family and comparison with other methods.

Input sequences:

8138 GH13 proteins downloaded from CAZy (www.cazy.org) on 01.Sep.2011.

5 Algorithm and implementation:

The implementation of PPR where each protein was used to generate a group of protein in each repetition of the algorithm (example 9) is computational intensive. To reduce the amount of computation and reach similar results we developed the following procedure:

- 10 1. Generate a list of all n-mer peptides in all the proteins and a list of peptides for each protein.
2. Remove all the peptides that occur in only one protein.

This step removes all peptides that occur in only one protein from the subsequent calculations.

15 For each protein used as seed protein:

1. Make all the n-mer peptides that occur in the sequence of the seed protein.
 2. Count the number of proteins that contain more than a cut-off value of the peptides.
 3. Assign this number to the seed protein.
- 20 All the seed proteins were ranked according to this number with the seed protein with the highest number first. Next, the 100 highest ranked proteins on this list were used as seed proteins for PPR analysis and the largest group of proteins that came out of the analysis was selected as a subfamily. These proteins were removed from the list of proteins and from the list of seed proteins before repeating the PPR analysis. This
- 25 step significantly reduced the number of calculations when many proteins were used as input as only 100 seed proteins were used in each round of PPR instead of using all the proteins.

Results:

The glycoside hydrolase family 13 (GH13) proteins have previously been classified into subfamilies ((Stam et al. 2006), www.cazy.org). PPR analysis with the parameters that worked best for GH5 classification in example 9 (peptide length = 6, 70 peptides, cut off = 10, stringency 0.14) of 8138 GH13 proteins downloaded from CAZy yielded 5 50 subfamilies. Cross comparison of 5195 proteins that were assigned to both a CAZy subfamily ((Stam et al. 2006), www.cazy.org) and a PPR subfamily showed that on average 89 % of the proteins in each CAZy subfamily belonged to one PPR subfamily and vice versa. Only 40 PPR subfamilies overlapped with CAZy subfamilies and for 31 of these 40 PPR subfamilies all the proteins belonged to one CAZy subfamily. The 10 analysis performed by Stam and coworkers (Stam et al. 2006) implied the sequential use of several algorithms for sequence alignment, clustering (SECATOR) and removal of sequences with insufficient similarity with other sequences. Although only 1691 GH13 sequences were present in CAZy at the time of analysis, the authors report that the method was very time consuming (Stam et al. 2006) and would be difficult to 15 perform at the time of publication when the number of GH13 sequences in CAZy had grown to 2456. PPR analysis of 8138 GH13 took 7 hours with a script written in Ruby, which is a relatively slow programming language, and runned on a powerful desktop computer (IntelR CoreTm i7-2600 CPU @ 3.40 GHz; 16 GB RAM). It took less than 25 minutes to perform a PPR analysis of 1691 sequences chosen at random from the 20 8138 GH13.

By assigning 204 GH13 proteins where the activity has been determined experimentally to subgroups it was found that 91 % of these proteins were classified to subfamilies with only one function.

Conclusion:

25 These results suggest that analysis of the GH13 proteins with PPR leads to largely the same results as found by considerably more tedious methods. Furthermore, PPR classified the GH13 proteins into functionally relevant subfamilies as was found for GH5 proteins (example 9).

Example 12. Use of Peptide Pattern Recognition-generated peptide lists for mining the 30 *Schizophyllum commune* genome.

Input sequence:

Schizophyllum commune whole genome sequence (accession number: 302674088).

5 Algorithm and implementation:

The *Schizophyllum commune* whole genome nucleotide sequence was divided into 2000 bases long fragments. This was done once by starting at base number 1 and once starting at base number 1000 thus generating two sets of fragments with 1000 bases overlap.

- 10 Each fragment was reverse translated in all three reading frames on one strand to generate one set of possible open reading frames (forward reading frames), and in all three reading frames on the other strand to generate the other set of possible open reading frames (reverse reading frames).

Next, all reading frames were scored against the subfamily -specific peptide lists
15 generated in example 9 by:

1. Finding all the peptides from the list that were present in the sequence of the reading frame.
2. Sum the frequency of these peptides. This gave the subfamily-specific frequency score.

20

Finally, the score and the number of hits for all three forward reading frames, respectively for all three reverse reading frames was summarized and reported for all sequences that contained more than two of the peptides from the GH5 subfamilies (example 9).

25 Results:

Screening of the *Schizophyllum commune* genome with GH5 subfamilies-specific peptide lists yielded 18 sequences that could be assembled into nine overlapping

sequences. The open reading frame with the highest subfamily-specific frequency score was used for a BLAST search at <http://blast.ncbi.nlm.nih.gov/>. Eight of the sequences (accession numbers: 302683907, 302684769, 302679542, 302698407, 302698219, 302695291, 302689835 and 302687152) were annotated as GH5 proteins
5 in the protein sequence database whereas the last protein accession number: 302698617) was annotated as unknown but shows identity to P450 enzymes and is probably a false positive i.e. not a GH5 protein.

By scoring the eight annotated GH5 proteins against the subfamily -specific peptide lists (see example 9) they were assigned as 2 cellulases (EC 3.2.1.4), 5 glucan 1,3- β -
10 glucosidases (EC 3.2.1.58) and 1 mannan endo- β -1,4-mannosidase (EC 3.2.1.78).

The only of these proteins where the activity has been investigated experimentally is the protein with accession number 302683907, which was predicted by Peptide Pattern Recognition to be a cellulase (EC 3.2.1.4) in agreement with the experimental evidence (Clarke et al. 1997).

15

Conclusion:

Peptide lists generated with Peptide Pattern Recognition can be used for searching genomes and fragments of genomes or other sequences for open reading frames encoding proteins resembling the peptide lists. This is a fast method to find new family
20 members and designate subfamilies and predict the function of the proteins encoded by the nucleotide sequences.

25

WE CLAIM:

1. A method of providing a library of n-mer sequences, the method comprising the steps of:
 - 5 (i) select one or more biopolymer sequence(s) from a specified group of biopolymers having a mixture of different biopolymers,
 - (ii) generate all possible n-mers from the sequence(s) wherein n is an integer of at least 2, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency
 - 10 to the n-mer,
 - (iii) use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s),
 - 15 (iv) generate all possible n-mers from the first group, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,
 - (v) optionally use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a second group of one or more
 - 20 sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences,
 - (vi) optionally repeat steps (iv) and (v) until no further biopolymers of the specified group of biopolymers are retrieved, and a definite number of biopolymers are identified, and
 - 25 (vii) group all provided n-mers from the biopolymers into the library of n-mer sequences.
2. The method of claim 1 for generating one or more selected primers or probes useful for the detection of at least one sequence from a nucleic acid sample, the
- 30 method comprising the steps of:
 - in step (iv) generate all possible n-mers from the first group and for each n-mer sequence calculate the frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,

and in step (viii) decide whether the n-mers are peptides or nucleic acids, if the n-mers are peptides reverse translate the n-mers into a corresponding nucleic acid consensus sequence, and provide at least one nucleic acid consensus sequence from one of the most frequent occurring n-mers, and define whether it is a probe or a primer.

5

3. The method of claim 2, wherein two primers are provided separated by a suitable distance and, wherein the reverse primer is complementary to the nucleic acid consensus sequence.

10 4. The method of any one of claims 2-3, wherein the nucleic acid consensus sequence of step (iv) is translated according to the genetic code.

5. The method of any one of claims 2-4, wherein primer set further is selected to have a desired redundancy.

15

6. A method of providing a library of biopolymer sequences having one or more n-mers in common, wherein the library is composed of at least 2 biopolymer sequences, the method comprising the steps of:

(i) select one or more biopolymer sequence(s) from a specified group of biopolymers
20 having a mixture of different biopolymers,

(ii) generate all possible n-mers from the sequence(s) wherein n is an integer of at least 2, optionally for each n-mer sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,

25 (iii) use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a first group, from the specified group of biopolymers, of 2 or more biopolymers having one or more of the n-mer(s) in the biopolymer sequence(s),

(iv) generate all possible n-mers from the first group, optionally for each n-mer
30 sequence calculate frequency as the number of biopolymers that contain the n-mer sequence at least once and assign the frequency to the n-mer,

(v) use the generated n-mers or a selection of the generated n-mers according to a specified frequency to provide a second group of one or more sequences from the specified group of biopolymers having one or more of the n-mer(s) in the sequences,
(vi) repeat steps (iv) and (v) until no further biopolymers of the specified group of
5 biopolymers are retrieved, and a definite number of biopolymers are identified,
(vii) group all provided biopolymers into the library of biopolymer sequences having one or more of the n-mer(s).

7. The method of any one of claims 1-6 wherein the mixture of different
10 biopolymers are from a predefined source, such as a gene bank, a known micro organism, sample, protein, gene family, chromosomes from one organism or selection of chromosome sequences or parts hereof from several organisms or from an unknown source such as a new micro organism, a pool of unidentified biopolymers, or a mixture of known and unknown sources, an environmental sample e.g. from a
15 mammal, microorganism, plant sample, mixture of organisms, sample of unordered sequence reads e.g., from one or several organisms, or from a database.

8. The method of any one of claims 1-7 wherein n is an integer from 2 to 75.

20 9. The method of any one of claims 6-8 wherein the library of biopolymer sequences has sequence similarity of at least 1% identity, or has the same bioactivity, or is from the same gene family, or from the same type or phylogenetic class of organisms, from the same organism, from the same sample containing biopolymers from one or more organisms.

25

10. The method of any one of claims 1-9 wherein the biopolymer sequence is selected from polypeptides, proteins, nucleic acids, or fragments thereof.

11. The method of any one of claims 1-10 wherein the n-mers are ranked according
30 to frequency and only the 90% most frequent n-mers are used in step (iii) or (v) or both.

12. The method of any one of claims 1-10 wherein 3 to 150 n-mers, such as 100, having the highest frequency are used in step (iii) or (v) or both.

13. The method of any one of claims 1-12 wherein the method is performed *in silico*.

5

14. Use of the selected primers and/or probes of any one of claims 2-5 or 7-13 for amplification of nucleic acids, as hybridization probes for screening a library of nucleotide sequences, as an expression library of peptides or polypeptides, both in wet lab and *in silico* screening.

10

15. Use of a library of n-mer sequence(s) obtainable by the method of claim 1 from a biopolymer sequence having an identified sequence for searching and collecting fragments of biopolymers from an environment comprising the biopolymer fragments, by *in silico* screening and collecting any biopolymer fragment(s) that has a sequence
15 identical to, complementary to or related by translation or reverse translation to the sequence of one or more of the selected primers and/or probes.

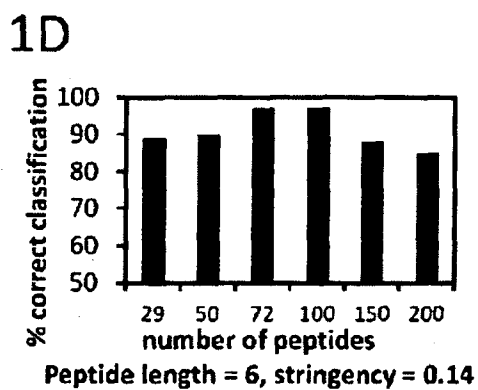
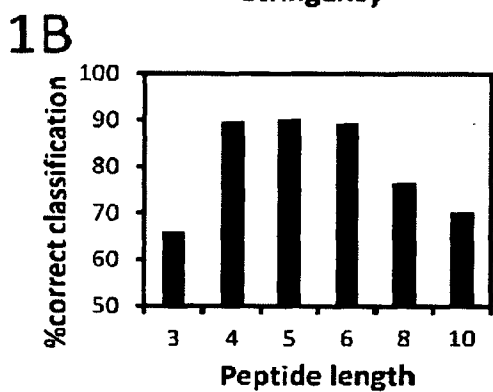
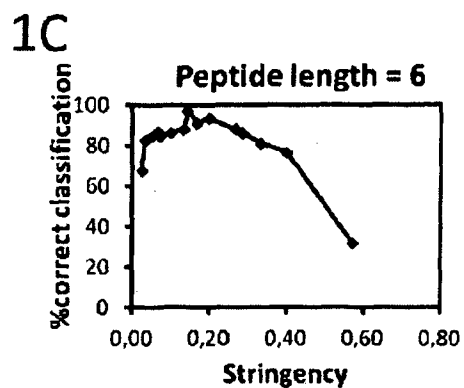
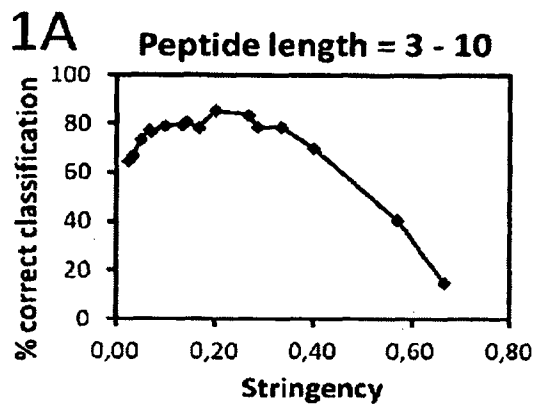
16. Use of a library of n-mer sequences generated from a library of one or more known biopolymer sequences having a known function in common to identify a different

20 biopolymer sequence comprising the n-mer sequences and having the same function as the known biopolymer sequence(s), by comparing the library of n-mer sequences of the known biopolymer sequence(s) with the n-mer sequences from the different biopolymer sequence(s), and selecting the different biopolymer sequence(s) having at least 1 of the n-mers from the library of n-mers of the known biopolymer sequence(s)
25 in common.

17. The use of claim 16 further comprising the step of screening the selected different biopolymer sequence(s) in a relevant assay to confirm that the function is the same as the known biopolymer sequence(s).

30

1/4



2A

Subfamily	Proteins
1	21 EC 3.2.1.4
2	16 EC 3.2.1.58, 1 EC 3.2.1.75
3	13 EC 3.2.1.4
4	11 EC 3.2.1.78
5	10 EC 3.2.1.4
6	9 EC 3.2.1.4
7	6 EC 3.2.1.78
8	6 EC 3.2.1.75
9	4 EC 3.2.1.78

2B

Subfamily	1	2	3	4	5	6	7	8	9
1	67	0	0	0	1	0	0	0	0
2	0	66	0	0	0	0	0	3	0
3	0	0	69	0	0	0	0	0	0
4	0	0	0	51	0	0	0	0	0
5	1	0	0	0	45	0	0	0	0
6	0	0	0	0	0	40	0	0	0
7	0	0	0	0	0	0	62	0	0
8	0	3	0	0	0	0	0	57	0
9	0	0	0	0	0	0	0	0	18

2C

Classification	Correct	Wrong
EC 3.2.1.4	59	1
EC 3.2.1.58	18	1
EC 3.2.1.75	7	0
EC 3.2.1.78	31	1
Total	115	3

2D

Subfamily	1	2	3	4	5	6	7	8	9
Σ frequency	29.18	0	0	0	0.8	0	0	0	0
Hits	52	0	0	0	1	0	0	0	0

3A

% correct predictions							
limit\pep_length	2	3	4	5	6	8	10
10	44	74	96	97	98	99	100
20	46	77	96	96	98	99	100
30	47	83	93	94	96	98	100
50	48	87	92	96	96	97	100
70	49	87	93	96	97	99	100
100	55	84	89	96	98	99	100
150	69	85	90	98	98	98	100
200	66	84	88	97	99	99	100
400	39	87	91	97	98	98	100
800		88	91	97	99	98	100
1600		87	92	98	98	98	100
3200		91	92	97	98	99	100

3B

% of proteins predicted							
limit\pep_length	2	3	4	5	6	8	10
10	67	92	67	48	40	34	26
20	83	96	69	61	48	41	31
30	89	97	73	60	53	44	34
50	94	98	77	66	58	43	37
70	96	98	86	73	58	51	38
100	98	98	87	75	62	52	43
150	98	98	92	79	69	54	40
200	98	98	94	79	73	53	45
400	98	98	95	85	77	63	46
800		98	97	90	80	63	47
1600		98	98	91	87	73	51
3200		98	98	92	86	74	57

4/4

3C

correct predictions * proteins predicted							
limit\pep_length	2	3	4	5	6	8	10
10	29	68	64	46	39	33	26
20	38	73	66	58	47	40	31
30	41	80	67	56	50	43	34
50	45	85	70	63	55	41	37
70	47	85	79	70	56	50	38
100	53	82	77	72	60	51	43
150	67	83	82	77	67	52	40
200	64	82	82	76	72	52	45
400	38	85	86	82	75	61	46
800		86	88	87	79	61	47
1600		85	90	89	85	71	51
3200		89	90	89	84	73	57

3D

peptide length	2	3	4	5	6	8	10
number of peptides	400	8×10^3	16×10^4	32×10^5	64×10^6	256×10^8	512×10^9
correct predictions * proteins predicted	38	91	92	89	88	79	75

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2012/051099

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F19/20
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GIUSEPPE D'AURIA ET AL: "IWoCS: analyzing ribosomal intergenic transcribed spacers configuration and taxonomic relationships", BIOINFORMATICS, vol. 22, no. 5, 10 January 2006 (2006-01-10), pages 527-531, XP055001789, DOI: 10.1093/bioinformatics/btk033 the whole document ----- -/--	1-17



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
 "&" document member of the same patent family

Date of the actual completion of the international search

28 March 2012

Date of mailing of the international search report

04/04/2012

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Kürten, Ivayla

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2012/051099

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	BETTY YEE MAN CHENG ET AL: "Protein Classification based on Text Document Classification techniques", PROTEINS: STRUCTURE, FUNCTION, AND BIOINFORMATICS, vol. 58, 11 January 2005 (2005-01-11), pages 955-970, XP055001791, DOI: 10.1002/prot.20373 abstract page 958 - page 960	1-17
X	----- JOHN K VRIES ET AL: "A Sequence Alignment-Independent Method for Protein Classification", APPLIED BIOINFORMATICS, vol. 3, no. 2-3, 1 January 2004 (2004-01-01), pages 137-148, XP055001794, abstract page 139 - page 143, left-hand column	1-17
X	----- TOMOVIC A ET AL: "n-Gram-based classification and unsupervised hierarchical clustering of genome sequences", COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE, ELSEVIER, AMSTERDAM, NL, vol. 81, no. 2, 1 February 2006 (2006-02-01), pages 137-153, XP024895751, ISSN: 0169-2607, DOI: DOI:10.1016/J.CMPB.2005.11.007 [retrieved on 2006-02-01] abstract page 138 - page 141 page 144, right-hand column - page 147	1-17
