2007

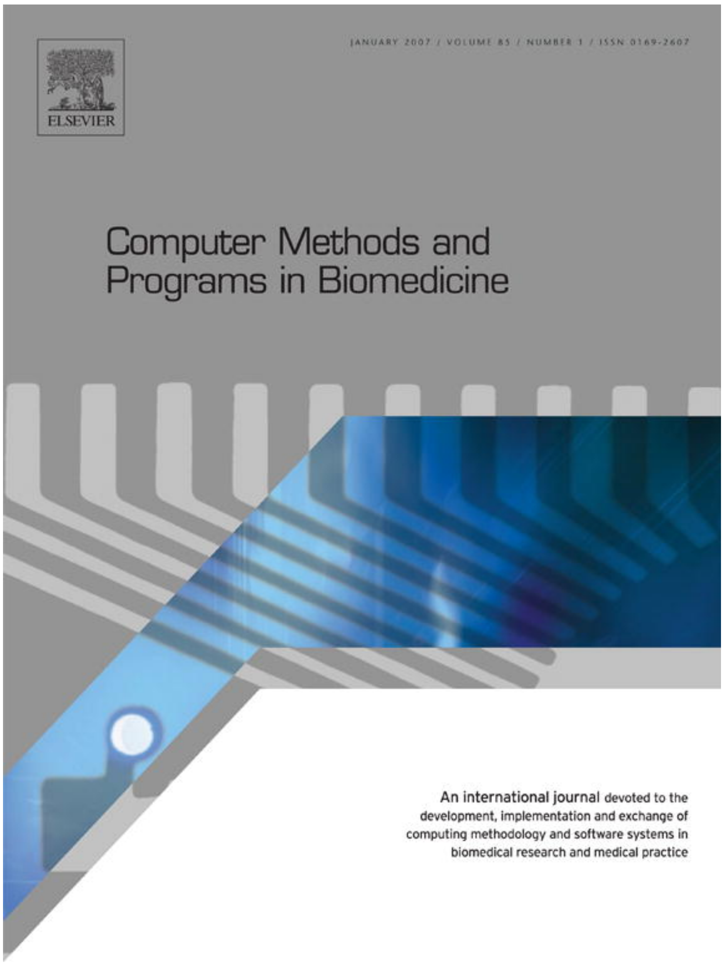
ProteinParser:a Community Based Tool for the Generation of a Detailed Protein Consensus and FASTA Output

Barry Ryan
Technological University Dublin, barry.ryan@tudublin.ie

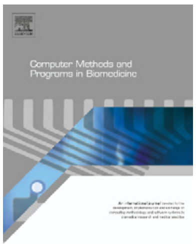
Ronan Barrett

Recommended Citation

Ryan, B.J. & Barrett, R. (2007). ProteinParser--a community based tool for the generation of a detailed protein consensus and FASTA output. *Computer Methods Programs Biomedicine,* 85, (1), 69-76. doi:10.1016/j.cmpb.2006.09.015


JANUARY 2007 / VOLUME 85 / NUMBER 1 / ISSN 0169-2607
ELSEVIER
Computer Methods and Programs in Biomedicine
An international journal devoted to the development, implementation and exchange of computing methodology and software systems in biomedical research and medical practice

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



ProteinParser—A community based tool for the generation of a detailed protein consensus and FASTA output

Barry J. Ryan[a,*], *Ronan Barrett*[b]

[a] *School of Biotechnology and National Centre for Sensor Research, Dublin City University, Dublin 9, Ireland*
[b] *School of Computing, Dublin City University, Dublin 9, Ireland*

ARTICLE INFO

Article history:
Received 4 May 2006
Received in revised form
28 September 2006
Accepted 29 September 2006

Keywords:
Protein alignment
Consensus concept
Amino acid frequency counting
Protein stabilisation

ABSTRACT

Comparison of bioinformatic data is a common application in the life sciences and beyond. In this communication, a novel Java based software tool, *ProteinParser*, is outlined. This software tool calculates a detailed consensus, or most common, amino acid at a given position in an aligned protein set, whilst also generating a full consensus protein FASTA output. A second application of this software tool, computing a consensus amino acid given a tolerance threshold, is also demonstrated. The phytase and the common bacterial β-lactamase proteins are analysed as 'proof of concept' examples. Consensus proteins, as generated by *ProteinParser*, are regularly utilised in the selection of residues for protein stabilisation mutagenesis; however, this widely applicable software tool will find many alternative applications in areas such as protein homology modelling.

1. Introduction

Comparison is one of the most basic procedures in biology. Measuring similarities and differences between organisms allows scientists to generate groups and clusters, from which we can infer such things as evolutionary relationships [1]. Simultaneous alignment of biological sequences, nucleotide or amino acid, allows rapid detection of homology between seemingly unrelated proteins, however, continual development of bioinformatic software capable of processing vast data sets is paramount [2–4]. The interested reader is directed to the recent review in this area [5]. One application of such alignment procedures is in the identification of key stabilising residues in a protein structure, utilising the Consensus Approach [6]. This *semi-rational* protein design methodology predicts potential stabilising mutations *in silico*, which can be validated experimentally by site directed mutagenesis. Several reports of successful protein stabilisation employing this approach are noted in the literature [7–9]. The consensus approach is based on the assumption that conserved residues, as detailed from sequence alignments of related proteins, contribute more to the stabilisation of the protein than non-conserved residues [10]. It has been proven that a set of amino acid sequences of homologous, mesophilic enzymes contains sufficient information to allow rapid design of a thermostabilised, fully functional enzyme [11].

Based on these assumptions, a Java based sequence analysis program, "*ProteinParser*", was generated which can calculate consensus residues at any particular position in an aligned protein sequence. This software tool acts as a direct *add-on* to the commonly used Clustal W package [12], in which variables, such as gap penalty, etc., can be altered. The "*in silico*" consensus protein, in conjunction with crystal structure analysis, can be utilised to select key residues for stabilisation mutation, postulate evolutionary divergence or model homologous protein activity. In this communication, the phytase and β-lactamase proteins are analysed as an example of the potential applications of this novel bioinformatic software tool.

[*] *Corresponding author*. Tel.: +353 1 700 5470; fax: +353 1 700 5412.
E-mail address: Barry.Ryan5@mail.dcu.ie (B.J. Ryan).

0169-2607/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2006.09.015

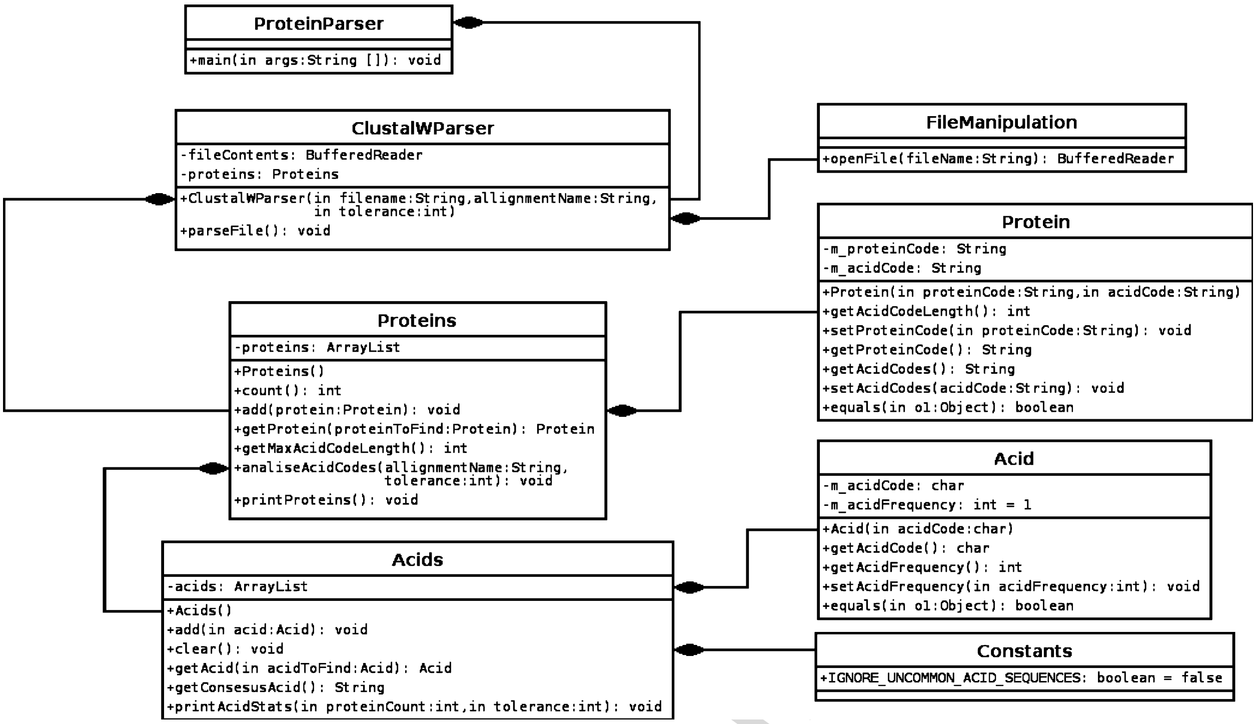

ProteinParser
+main(in args:String []): void
ClustalWParser
-fileContents: BufferedReader
-proteins: Proteins
+ClustalWParser(in filename:String,allignmentName:String, in tolerance:int)
+parseFile(): void
FileManipulation
+openFile(fileName:String): BufferedReader
Protein
-m_proteinCode: String
-m_acidCode: String
+Protein(in proteinCode:String,in acidCode:String)
+getAcidCodeLength(): int
+setProteinCode(in proteinCode:String): void
+getProteinCode(): String
+getAcidCodes(): String
+setAcidCodes(acidCode:String): void
+equals(in o1:Object): boolean
Proteins
-proteins: ArrayList
+Proteins()
+count(): int
+add(protein:Protein): void
+getProtein(proteinToFind:Protein): Protein
+getMaxAcidCodeLength(): int
+analiseAcidCodes(allignmentName:String, tolerance:int): void
+printProteins(): void
Acid
-m_acidCode: char
-m_acidFrequency: int = 1
+Acid(in acidCode:char)
+getAcidCode(): char
+getAcidFrequency(): int
+setAcidFrequency(in acidFrequency:int): void
+equals(in o1:Object): boolean
Acids
-acids: ArrayList
+Acids()
+add(in acid:Acid): void
+clear(): void
+getAcid(in acidToFind:Acid): Acid
+getConsesusAcid(): String
+printAcidStats(in proteinCount:int,in tolerance:int): void
Constants
+IGNORE_UNCOMMON_ACID_SEQUENCES: boolean = false

Fig. 1 – Unified Modelling Language Class diagram of ProteinParser tool.

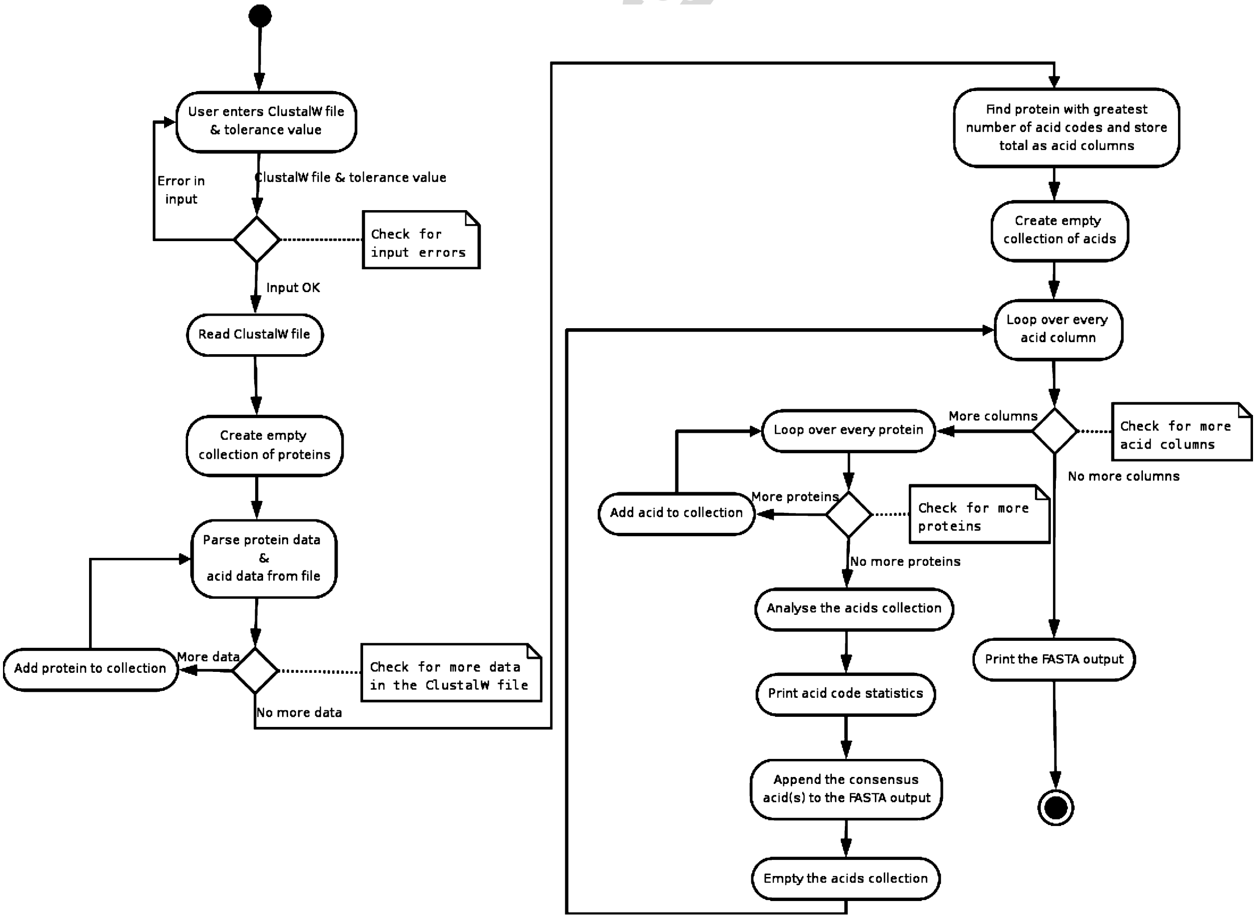

User enters ClustalW file & tolerance value
Error in input
ClustalW file & tolerance value
Check for input errors
Input OK
Read ClustalW file
Create empty collection of proteins
Parse protein data & acid data from file
Add protein to collection
More data
Check for more data in the ClustalW file
No more data
Find protein with greatest number of acid codes and store total as acid columns
Create empty collection of acids
Loop over every acid column
More columns
Check for more acid columns
No more columns
Loop over every protein
Add acid to collection
More proteins
Check for more proteins
No more proteins
Analyse the acids collection
Print acid code statistics
Append the consensus acid(s) to the FASTA output
Empty the acids collection
Print the FASTA output

Fig. 2 – Unified Modelling Language workflow diagram for ProteinParser tool.

2. Software development and application

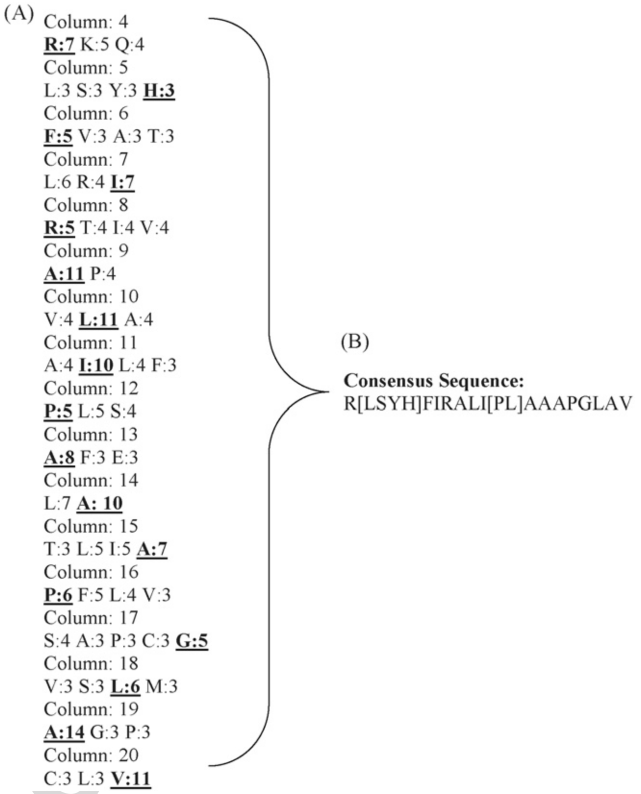
ProteinParser is a novel command line based software tool for automatically producing reports on the amino acid frequency within protein alignments. The tool is written in Java, a platform neutral language, and requires only the Java 2 runtime environment for execution. The tool utilises Object Oriented Software Development [13] mechanisms to modularise the codebase, making the tool easily extensible. There are eight classes, as shown is Fig. 1. The function of each class is outlined in Table 1. A workflow diagram, Fig. 2, and the following discussion illustrate how the tool works.

A ClustalW file [12], a suitable consensus name and a tolerance value between 0 and 100 are taken as program inputs. The tolerance value dictates how often an amino acid must be present, at a particular position, within a number of proteins before its frequency is reported by the tool. The tool works by first reading a valid ClustalW file (parseFile method of ClustalWParser class). The file is analysed and all the unique protein names are recorded and represented in the tool by Protein classes in a Proteins container class. The acid codes for each protein are then parsed from the ClustalW file, resulting in the creation of an acid code string representing each of the acids in a protein.

The detailed consensus data is obtained dynamically by calculating which protein has the highest number of acid codes (getMaxAcidCodeLength method of Proteins class). This value allows direct line up all the proteins for a comparative analysis. A given alignment position can then be looped over for each protein (analiseAcidCodes method in Proteins class), termed columns in the tool, and an Acid instance created to represent each discrete acid code. These Acid instances are stored in an Acids collection to allow for easier calculation of the consensus. Consensus values are computed by incrementing the frequency of a given acid each time it is found at a given alignment position. Detailed consensus information for each alignment position is reported if the consensus is above the tolerance value entered by the user when starting up the tool (printAcidStats method of Acids class). The tolerance is evaluated by taking the frequency of an acid in a given position and dividing this value by 100 over the tolerance value inputted.

Table 1 – Details of ProteinParser classes

Class	Description
ProteinParser	Entry point to tool. Takes ClustalW file, candidate alignment name and tolerance values as input
ClustalWParser	Parses ClustalW file
FileManipulation	Generic file access functionality
Proteins	Holder for Protein classes
Protein	Abstraction of a protein with some useful interrogation and manipulation methods
Acids	Holder for amino acid classes
Acid	Abstraction of an amino acid with some useful interrogation and manipulation methods
Constants	Central location of constants used by tool


(A)

```
Column: 4
R:7 K:5 Q:4
Column: 5
L:3 S:3 Y:3 H:3
Column: 6
F:5 V:3 A:3 T:3
Column: 7
L:6 R:4 I:7
Column: 8
R:5 T:4 I:4 V:4
Column: 9
A:11 P:4
Column: 10
V:4 L:11 A:4
Column: 11
A:4 I:10 L:4 F:3
Column: 12
P:5 L:5 S:4
Column: 13
A:8 F:3 E:3
Column: 14
L:7 A: 10
Column: 15
T:3 L:5 I:5 A:7
Column: 16
P:6 F:5 L:4 V:3
Column: 17
S:4 A:3 P:3 C:3 G:5
Column: 18
V:3 S:3 L:6 M:3
Column: 19
A:14 G:3 P:3
Column: 20
C:3 L:3 V:11
```

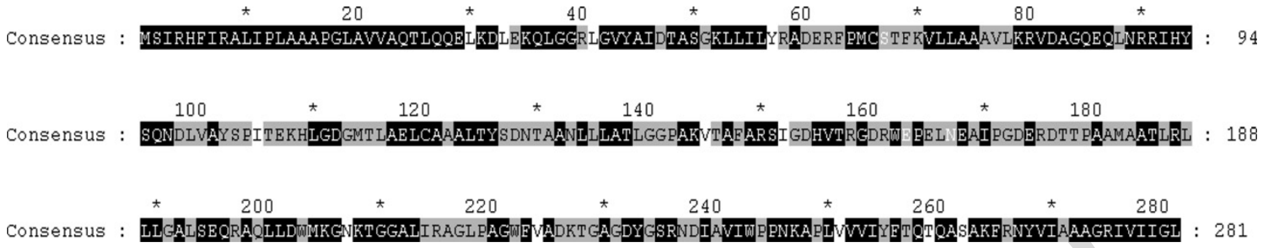
(B)

```
Consensus Sequence:
R[LSYH]FIRALI[PL]AAAPGLAV
```

Fig. 3 – Actual ProteinParser output, detailed amino acid frequency (greater than two occurrences) is shown in Part A. The tool selected the most frequent amino acid (emboldened and underlined in Part A) and generated a consensus protein (B). Where more than one consensus residue was found, all equal values were wrapped in square brackets. This region corresponds to Fig. 4, residues 4–20.

The tool, upon completion of the detailed consensus data reporting, also outputs a FASTA based consensus sequence. The method functions like the detailed consensus by incrementing the frequency of a given acid for each time it is found at a given alignment position, but only reports the most frequent acid(s) for the entire alignment as the FASTA output (getConsensusAcidCode method in Acids class). However, multiple amino acids may be equally frequent, this is noted in the tool output by square brackets around multiple consensus acids. It is assumed the end-user can choose the most appropriate consensus amino acid for a given alignment. Both the square brackets and the alternative consensus acids should be deleted by the end-user before importing the FASTA output to a FASTA compatible tool.

Genedoc (www.psc.edu/biomed/genedoc), an external stand-alone multiple sequence alignment editor and shading utility that provides extensive display and highlighting facilities, was utilised to view multiple aligned consensus sequences generated by the tool as ProteinParser output is not graphical (see Fig. 3). Graphical representations (Figs. 4–6) permitted rapid and simple assessment of ProteinParsers consensus output in comparison to alternative sequences. Sequences must be previously aligned for correct Genedoc



```
                      *        20         *        40         *        60         *        80         *
Consensus : MSIRHFIRALIPLAAAPGLAVVAQTLQQELKDLEKQLGGRLGVYAIDTASGKLLILYRADERFPMCSTFKVLLAAAVLKRVDAGQEQLNRRIHY :  94

               100         *       120         *       140         *       160         *       180
Consensus : SQNDLVAYSPITEKHLGDGMTLAELCAAALTYSDNTAANLLLATLGGPAKVTAFARSIGDHVTRGDRWEPELNEAIPGDERDTTPAAMAATLRL : 188

             *       200         *       220         *       240         *       260         *       280
Consensus : LLGALSEQRAQLLDWMKGNKTGGALIRAGLPAGWFVADKTGAGDYGSRNDIAVIWPPNKAPLVVVIYFTQTQASAKFRNYVIAAAGRIVIIGL : 281
```

Fig. 4 – *ProteinParser* generated basic Consensus β-lactamase Protein. The software produced the basic Consensus protein by calculating the most common residue at a particular position. The %consensus varied from 10% to 100%, with no attempt made to correlate data with previous scientific knowledge. White colouration indicates 10–25% consensus between the aligned Consensus proteins. Black colouration represents between 25% and 50% residue conservation throughout the aligned Consensus proteins, whereas grey colouration with black lettering indicates greater than 50% residue conservation throughout the aligned Consensus proteins. Catalytically important amino acids [24] are highlighted in white lettering on a grey background. Image was generated utilising the GeneDoc software package [23].

display and, as such, Clustal W's "Pileup" function of multiple sequences served as input data.

As proof of concept, the software tool was initially applied to the β-lactamase protein. The following search boolean was used to search the NCBI protein repository database.

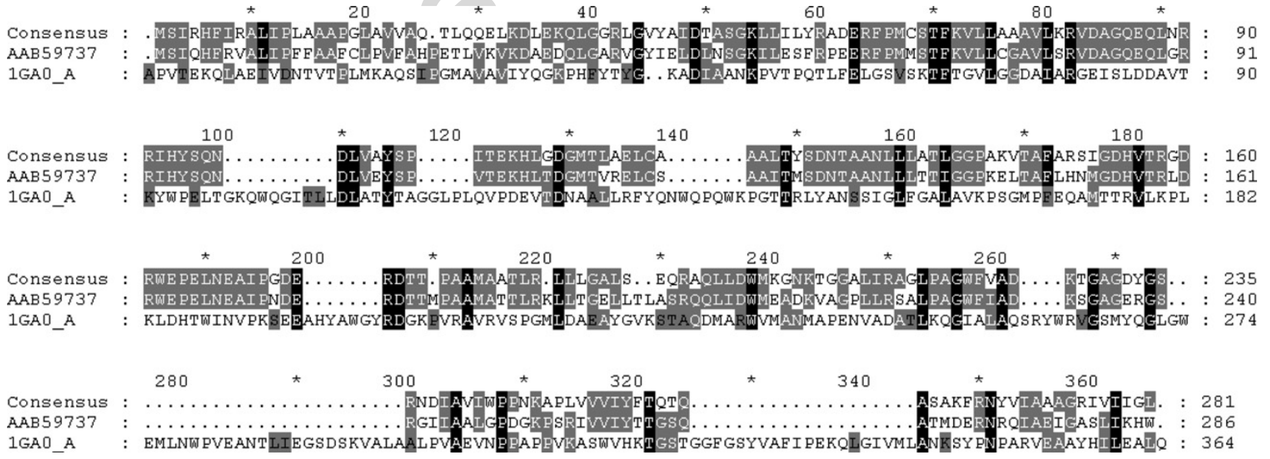
"((((((β-lactamase) AND (bacteria))) NOT (precursor)) NOT (putative)) NOT (Segment))"

Over 6000 results were obtained, of which 20 randomly chosen, unique, β-lactamase sequences are downloaded (see Table 2). These sequences were then aligned via the Clustal W alignment package, using the default alignment parameters for the Clustal W server (www.ebi.ac.uk/clustalw). The alignment was saved as a ".aln" file and subsequently processed by the "*ProteinParser*" software tool to generate a basic consensus sequence. The *ProteinParser* software tool also analysed the aligned sequences, applying a 50% tolerance in selecting the consensus amino acid.

The phytase protein was chosen as a second example of the potential application of *ProteinParser*. The sequences chosen for alignment in this case were previously outlined by Lehmann et al. [11]. A tolerance of "0" was set, and the percentage occurrence of each amino acid at each relative position was reported, providing detailed information on amino acid content. The FASTA output function of the tool then selected the most common amino acid and generated the overall consensus protein.

Table 2 – List of β-lactamase protein accession numbers (NCBI database) utilised in the generation of the basic Consensus β-lactamase Protein

BAA02563	CAA56427	AAA19882	CAA38522
CAB37325	CAA06311	CAB69042	AAA22904
P80545	Q8XDQ2	YP_209323	AAX86805
CAG25812	NP_052173	CAA88908	CAA37052
CAA79967	BAA14224	AAA24777	YP_337703
CAI43427	EAP41339	ZP_00859646	EAO98083



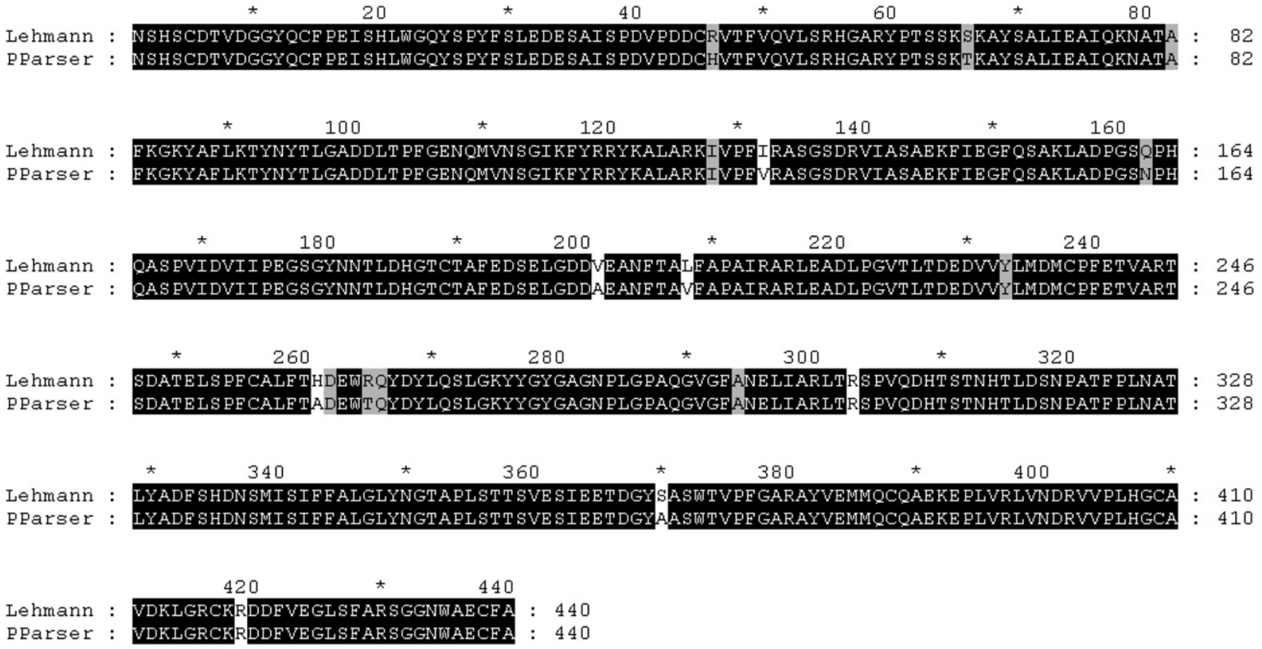
```
                    *        20         *        40         *        60         *        80         *
Consensus : .MSIRHFIRALIPLAAAPGLAVVAQ.TLQQELKDLEKQLGGRLGVYAIDTASGKLLILYRADERFPMCSTFKVLLAAAVLKRVDAGQEQLNR :  90
AAB59737  : .MSIQHFRVALIPFFAAFCLPVFAHPETLVKVKDAEDQLGARVGYIELDLNSGKILESFRPEERFPMMSTFKVLLCGAVLSRVDAGQEQLGR :  91
1GA0_A    : APVTEKQLAEIVDNTVTPLMKAQSIPGMAVAVIYQGKPHFYTYG..KADIAANKPVTPQTLFELGSVSKTFTGVLGGDAIARGEISLDDAVT :  90

                  100         *       120         *       140         *       160         *       180
Consensus : RIHYSQN..........DLVAYSP.....ITEKHLGDGMTLAELCA.......AALTYSDNTAANLLLATLGGPAKVTAFARSIGDHVTRGD : 160
AAB59737  : RIHYSQN..........DLVEYSP.....VTEKHLTDGMTVRELCS.......AAITMSDNTAANLLLTTIGGPKELTAFLHNMGDHVTRLD : 161
1GA0_A    : KYWPELTGKQWQGITLLDLATYTAGGLPLQVPDEVTDNAALLRFYQNWQPQWKPGTTRLYANSSIGLFGALAVKPSGMPFEQAMTTRVLKPL : 182

                *       200         *       220         *       240         *       260         *
Consensus : RWEPELNEAIPGDE.......RDTT.PAAMAATLR.LLLGALS..EQRAQLLDWMKGNKTGGALIRAGLPAGWFVAD....KTGAGDYGS.. : 235
AAB59737  : RWEPELNEAIPNDE.......RDTTMPAAMATTLRKLLTGELLTLASRQQLIDWMEADKVAGPLLRSALPAGWFIAD....KSGAGERGS.. : 240
1GA0_A    : KLDHTWINVPKSEEAHYAWGYRDGKPVRAVRVSPGMLDAEAYGVKSTAQDMARWVMANMAPENVADATLKQGIALAQSRYWRVGSMYQGLGW : 274

             280         *       300         *       320         *       340         *       360
Consensus : .......................RNDIAVIWPPNKAPLVVVIYFTQTQ....................ASAKFRNYVIAAAGRIVIIGL. : 281
AAB59737  : .......................RGIIAALGPDGKPSRIVVIYTTGSQ....................ATMDERNRQIAEIGASLIKHW. : 286
1GA0_A    : EMLNWPVEANTLIEGSDSKVALAALPVAEVNPPAPPVKASWVHKTGSTGGFGSYVAFIPEKQLGIVMLANKSYPNPARVEAAYHILEALQ : 364
```

Fig. 5 – GeneDoc comparison of the *ProteinParser* generated Consensus β-lactamase Protein, standard pBR322 β-lactamase protein (AAB59737) and *E. cloacae* β-lactamase (1GA0_A) [14] aligned via Clustal W [12]. Conserved residues between the Consensus β-lactamase, the standard β-lactamase and the *E. cloacae* β-lactamase are shaded in a variety of colours. Black colouration specifies residues that display 100% identity throughout the aligned Consensus proteins, grey with white lettering signifies consensus between two of the three sequences, and white with black lettering signifies no identity between the three β-lactamase proteins. Amin et al.'s [14] stabilising mutations are highlighted by black lettering on a grey background. Image was generated utilising the GeneDoc software package [23].



```
                      *        20         *        40         *        60         *        80
Lehmann : NSHSCDTVDGGYQCFPEISHLWGQYSPYFSLEDESAISPDVPDDCRVTFVQVLSRHGARYPTSSKSKAYSALIEAIQKNATA :  82
PParser : NSHSCDTVDGGYQCFPEISHLWGQYSPYFSLEDESAISPDVPDDCHVTFVQVLSRHGARYPTSSKTKAYSALIEAIQKNATA :  82

                    *       100         *       120         *       140         *       160
Lehmann : FKGKYAFLKTYNYTLGADDLTPFGENQMVNSGIKFYRRYKALARKIVPFIRASGSDRVIASAEKFIEGFQSAKLADPGSQPH : 164
PParser : FKGKYAFLKTYNYTLGADDLTPFGENQMVNSGIKFYRRYKALARKIVPFVRASGSDRVIASAEKFIEGFQSAKLADPGSNPH : 164

                  *       180         *       200         *       220         *       240
Lehmann : QASPVIDVIIPEGSGYNNTLDHGTCTAFEDSELGDDVEANFTALFAPAIRARLEADLPGVTLTDEDVVYLMDMCPFETVART : 246
PParser : QASPVIDVIIPEGSGYNNTLDHGTCTAFEDSELGDDAEANFTAVFAPAIRARLEADLPGVTLTDEDVVYLMDMCPFETVART : 246

                *       260         *       280         *       300         *       320
Lehmann : SDATELSPFCALFTHDEWRQYDYLQSLGKYYGYGAGNPLGPAQGVGFANELIARLTRSPVQDHTSTNHTLDSNPATFPLNAT : 328
PParser : SDATELSPFCALFTADEWTQYDYLQSLGKYYGYGAGNPLGPAQGVGFANELIARLTRSPVQDHTSTNHTLDSNPATFPLNAT : 328

              *       340         *       360         *       380         *       400         *
Lehmann : LYADFSHDNSMISIFFALGLYNGTAPLSTTSVESIEETDGYSASWTVPFGARAYVEMMQCQAEKEPLVRLVNDRVVPLHGCA : 410
PParser : LYADFSHDNSMISIFFALGLYNGTAPLSTTSVESIEETDGYAASWTVPFGARAYVEMMQCQAEKEPLVRLVNDRVVPLHGCA : 410

                  420         *       440
Lehmann : VDKLGRCKRDDFVEGLSFARSGGNWAECFA : 440
PParser : VDKLGRCKRDDFVEGLSFARSGGNWAECFA : 440
```

Fig. 6 – *ProteinParser* versus PRETTY [11] generated basic Consensus Phytase Proteins. Common amino acids predicted by both methodologies are highlighted with a black background. The PRETTY generated program failed to generate a consensus protein at 10 positions, highlighted by grey shading. For these positions, either the most frequent, or an arbitrarily chosen amino acid, was introduced. For eight additional positions, highlighted in white shading, specific phytase amino acids were chosen instead of the consensus amino acid [11]. Image was generated utilising the GeneDoc software package [23].

3. Results

3.1. *β-Lactamase example*

Recently, Amin et al. [14] generated an *Enterobacter cloacae* β-lactamase consensus protein by aligning 38 β-lactamase protein homologs deposited in Genbank. With this alignment, Amin identified 29 positions where the parental *E. cloacae* protein deviated from the consensus protein. Subsequent directed mutagenesis of these non-consensus residues, to consensus amino acids, produced a significantly thermostabilised β-lactamase protein. As proof of concept for the software tool discussed in this paper, 20 β-lactamase protein sequences, from various origins, were downloaded, aligned and interrogated by the *ProteinParser* program. Both applications of *ProteinParser*, basic consensus definition and tolerance analysis, were investigated utilising the β-lactamase protein as an example. Initially, to generate a basic Consensus β-lactamase Protein, 20 sequences were simply aligned and the most common amino acid at each position was calculated. This action produced a basic consensus model protein. A section of the outputted detailed consensus data and the corresponding consensus sequence, as generated by the tool, is detailed in Fig. 3. This output is not graphical and, as such, an external tool (Genedoc) was employed for comparative assessment and illustrative purposes. No attempt was made to correlate consensus amino acids with previous scientific knowledge. The *ProteinParser* software tool was also used to interrogate the Clustal W aligned sequences utilising a tolerance threshold of 50% conservation between the 20 proteins. The aligned proteins display 25% conservation above this threshold limit, as indicated in Fig. 4. Finally, standard β-lactamase (derived from the pBR322 cloning vector) and Amin's *E. cloacae* β-lactamase protein were aligned against the *ProteinParser* generated β-lactamase protein, and conserved regions highlighted (see Fig. 5). Interestingly, comparison of the *ProteinParser* Consensus β-lactamase Protein with the standard β-lactamase reveals an almost 70% conservation of amino acid residues. This compares with just 22% conservation with Amin et al.'s [14] β-lactamase. Although the reasoning behind this wide variation is beyond the scope of this report, it is clear that the choice of initial data, i.e. the aligned sequences, is critical to the generation of an accurate consensus protein. A BLAST search was utilised to select the initial homologs for Amin and co-workers alignment incorporating in Vector NTI software [15], as compared to a boolean-based search in this study. The divergence of the initial input sequence data for *ProteinParser*, in which bacterial as opposed to *E. cloacae* specific β-lactamase data were utilised, caused the large identity variation upon consensus comparison.

3.2. *Phytase example*

The phytase protein was also analysed as another example of the tools application. A consensus sequence, using available input data, has previously been published for this protein [11]. This allowed for direct comparison between *ProteinParser* and the PRETTY software tool utilised by Lehmann and co-workers. A "0" tolerance level was set within the *ProteinParser* software tool for the analysis of the Clustal W derived phytase alignment, with the most common amino acid chosen as the consensus amino acid at that position. The *ProteinParser* generated consensus protein displayed total agreement with the PRETTY consensus protein, with the exception of the residues (18) manually altered within Lehmann's published consensus [11]. Additionally, Lehmann et al. [11] arbitrarily selected several residues for manipulation and these could not have been predicted by ProteinParser (see Fig. 6).

4. Discussion

The consensus approach, in its simplest application, is a comparative methodology. Different relations of a particular protein are aligned against each other and the consensus, or most common, amino acid at a particular position is calculated [11]. However, in recent years, variations of this simple methodology have been implemented to stabilise β-lactamase [14] and to alter the cofactor specificity of a lactate dehydrogenase [9]. *In vivo*, proteins are under no selective pressure to form optimally stable structures; instead they tend to form structures of adequate stability, i.e. the protein is just stable enough not to limit the host organism's viability [16]. As such, there is a wide scope for stabilisation of mesophilic proteins, with substitution of non-consensus amino acids by consensus amino acids offering a feasible approach to improve the stability of a protein.

By applying a consensus-based approach, substitute amino acids can be selected simply and rapidly, at low cost and often resulting in the generation of a stabilised protein ([8], and references within). Quite simply, the "*in silico*" consensus protein dictates the replacement residues based on direct comparison between amino acids that deviate from the consensus protein. The first publication of consensus-based stabilisation was noted in 1989, with Pantoliano and co-workers increasing the unfolding temperature of Subtilisin BPN′ by six single amino acid substitution (N218S, G169A, Y217K, M50F, Q206C and N76D). All single substitutions resulted in increased thermal stability, with the combined six-mutant Subtilisin BPN′ molecule displaying an additive stabilisation effect [17]. The advantages of using this semi-rational selection procedure include the fact that the replacement amino acid has already proven its evolutionary fitness at that position, hence reducing the chances of incorporating a deleterious mutation, and no high-throughput selection procedure is required to select for improved mutants. Although no crystal structure is required, the availability of one is advantageous. However, several researchers have noted that stability differences between homologous proteins may be due to a very few naturally occurring sequence variations ([18], and references within). Magliery and Regan [16] have recently developed a more advanced consensus-based model, which accounts for some of the inadequacies of the basic consensus approach [11], including (i) accounting for regions of poor consensus and (ii) regions of high homology that mask subfamilies. Also, a collection of subfamilies may also negatively affect the basic consensus result, and this is rationally integrated into the statistical free energy model. This statistical free energy model does, however, require additional computation, global propensities are allocated to each amino acid allowing for the approximation of the statistical free energy for each position as a function of the binomial probability of a particular amino acid at any given position [16]. Magliery and Regan [16] addressed some of the major issues with the simple consensus approach; however, another important consideration is that input sequences are often unordered in terms of phylogenetic history, resulting in inaccurate consensus protein prediction. To overcome this problem, members from each branch of a phylogenetic tree could be utilised as input data. Recent advances in ancestral protein reconstruction may also allow the development of an ancestral protein consensus, permitting the evolutionary change in proteins to be followed '*in silico*', and ultimately predict potential stabilising mutations [19,20]. Another possible bioinformatic advance in consensus generation would be to incorporate additional amino acid information, such as physicochemical properties, allowing the generation of a general consensus sequence. This approach would result in fewer gaps in the consensus sequence and could also be included in combination with the ancestral protein consensus methodology.

Several other bioinformatic sequence alignment tools are freely available for download or use, including CINEMA [21] "PoPMusic" [22], GOCore (www.helsinki.fi/project/ritvos/GoCore) all that allow protein sequence alignment and analysis. CINEMA (Colour INteractive Editor for Multiple Alignments) is a very user-friendly package that allows visualisation and manipulation of both protein and DNA sequences, however no consensus predictions are available. PoPMusic (Prediction of Protein Mutations Stability Changes) is web-server based algorithm that predicts potentially stabilising mutations based on either "*in silico*" thermodynamic stability or predicted changes in free energy of folding. This method offers an alternative methodology to the consensus approach, it is however most accurate when limited to surface exposed residues predictions. It also requires a protein structure to carry out the algorithm, which may not be available in all cases. GoCore is a free to download Microsoft Excel^{TM} based plug-in that allows the user to generate useful visualisations of protein alignments. The 'conservation analysis menu' graphically displays residue conservation across an inputted species list. Inputted sequences can be aligned via T-Coffee [3] or by the GoCore software, however no specific consensus function is available.

Specific consensus generating software is both non- and commercially available, PRETTY (www.accelrys.com), Vector NTI (http://www.invitrogen.com), PHRAP (www.codoncode.com), Consensus Maker I (http://hiv-web.lanl.gov) and Consensus Maker II (http://web.comlab.ox.ac.uk/), all that have their relative advantages and disadvantages. PRETTY displays multiple sequence alignments and calculates a consensus sequence for realigned sequences; however, no detailed frequency information is generated, the output is in non-FASTA format and the software is not free-to-download. Vector NTI is routinely used for desktop sequence analysis and molecular biology data management. Although it provides a highly integrated application, combining all aspects of molecular biology, it is not free-to-download and does not provide a specific consensus calculating function. PHRAP is most commonly utilised as a leading program for DNA sequence assembly, i.e. generating a consensus DNA sequence for projects such as genome sequencing. The 'free to download' nature of this software tool is only applicable to academic users. Consensus Maker I is a free-to-use web based program consisting of two applications, a simple and an advanced consensus calculator. The basic calculator computes a consensus using customary default parameter choices. However, more parameters can be varied in the advanced calculator. The output data is not in FASTA format. Consensus Maker II is a simple Java based alignment tool for sequences that allows the user to build up a consensus sequence from a collection of input sequences. The user is required to paste the sequences sequentially, which is not suitable for large-scale sequence analysis. Also, no sequence alignment parameters can be varied within the software.

5. Conclusion

In conclusion, a simple, rapid and user-friendly Java based software tool has been developed that allows direct conversion of aligned protein sequences into a consensus protein. The program can generate a detailed, most frequent amino acid, consensus protein; along with applying a tolerance level to select only highly conserved residues. The consensus protein is outputted as a FASTA file, which allows continued bioinformatic analysis with other FASTA compatible software. This program acts as an add on to pre-existing, commonly used, software, in which many variables can be altered, gap penalty, etc. The primary function of this novel free-to-download software is in the selection of residues for protein stabilisation mutagenesis, as outlined in this communication. However, it is envisaged that this software will find many alternative applications, including modelling of novel and homologous proteins to elucidate possible enzymatic roles and evolutionary relationships of related proteins.

6. Mode of availability

The *ProteinParser* executable, source code and documentation is freely available for download at http://www.computing.dcu.ie/~rbarrett/Clusters/doc. The source code is open source and licensed under the GNU General Public License Version 2.

Acknowledgements

B.J.R. and R.B. thank the Irish Research Council for Science, Engineering and Technology for postgraduate scholarships, under the Embark Initiative, and the Research Advisory Panel of Dublin City University (Postgraduate Accommodation Scholarship, B.J.R.). Drs. Ciarán Ó'Fágáin, Mary J. O'Connell (School of Biotechnology, DCU) and the anonymous reviewers are also thanked for their constructive comments and critical review of the manuscript.

REFERENCES

[1] M. Levitt, M. Gerstein, A unified statistical framework for sequence comparison and structure comparison, Proc. Natl. Acad. Sci. U.S.A. 95 (1998) 5913–5920.

[2] R. Chenna, H. Sugawara, T. Koike, R. Lopez, T.J. Gibson, D.G. Higgins, J.D. Thompson, Multiple sequence alignment with the Clustal series of programs, Nucl. Acids Res. 31 (2003) 3497–3500.

[3] C. Notredame, D. Higgins, J. Heringa, T-Coffee: a novel method for multiple sequence alignments, J. Mol. Biol. 302 (2000) 205–217.

[4] F. Corpet, Multiple sequence alignment with hierarchical clustering, Nucl. Acids Res. 16 (1988) 10881–10890.

[5] T. Lassmann, E.L.L. Sonnhammer, Quality assessment of multiple alignment programs, FEBS Lett. 529 (2002) 126–130.

[6] M. Lehmann, M. Wyss, Engineering proteins for thermostability: the use of sequence alignments versus rational design and directed evolution, Curr. Opin. Biotechnol. 12 (2001) 371–375.

[7] J. Miyazaki, S. Nakaya, T. Suzuki, M. Tamakoshi, T. Oshima, A. Yamagishi, Ancestral residues stabilising 3-isopropylmalate dehydrogenase of an extreme thermopile: experimental evidence supporting the thermophilic common ancestor hypothesis, J. Biochem. 129 (2001) 777–782.

[8] M. Lehmann, C. Loch, A. Middendorf, D. Studer, S.F. Lassen, L. Pasamontes, A.P.G.M. van Loon, M. Wyss, The consensus concept for thermostability engineering of proteins: further proof of concept, Protein Eng. 15 (2002) 403–411.

[9] H. Flores, A.D. Ellington, A modified consensus approach to mutagenesis inverts the co-factor specificity of *Bacillus stearothermophilus* lactate dehydrogenase, Protein Eng. Des. Sel. 18 (2005) 369–377.

[10] B. van den Burg, V.G.H. Eijsink, Selection for increased protein stability, Curr. Opin. Biotechnol. 13 (2002) 333–337.

[11] M. Lehmann, L. Pasamontes, S.F. Lassen, M. Wyss, The consensus concept for thermostability engineering of proteins, Biochim. Biophys. Acta 1543 (2000) 408–415.

[12] J.D. Thompson, D.G. Higgins, T.J. Gibson, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucl. Acids Res. 22 (1994) 4673–4680.

[13] T.C. Lethbridge, R. Laganiere, Object-OrientedSoftware Engineering: Practical Software Development Using UML and Java, second ed., McGraw-Hill, New York, 2004.

[14] N. Amin, A.D. Liu, S. Ramer, W. Aehle, D. Meijer, M. Metin, S. Wong, P. Gualfetti, V. Schellenberger, Construction of stabilised proteins by combinatorial consensus mutagenesis, Protein Eng. Des. Sel. 17 (2004) 787–793.

[15] Schellenberger, Personal Communication, 16 January 2006.

[16] T.J. Magliery, L. Regan, Beyond consensus: statistical free energies reveal hidden interactions in the design of TPR Motif, J. Mol. Biol. 343 (2004) 731–745.

[17] M.W. Pantoliano, M. Whitlow, J.F. Wood, S.W. Dodd, K.D. Hardman, M.L. Rollence, P.N. Bryan, Large increases in general stability for Subtilisin BPN′ through incremental changes in free energy of unfolding, Biochemistry 28 (1989) 7205–7213.

[18] V.G.H. Eijsink, A. Bjork, S. Gaseidnes, R. Sirevag, B. Synstad, B. van der Berg, G. Vriend, Rational design of enzyme stability, J. Biotechnol. 113 (2004) 105–120.

[19] B.G. Hall, Simple and accurate estimation of ancestral protein sequences, Proc. Natl. Acad. Sci. U.S.A. 103 (2006) 5431–5436.

[20] B.S.W. Chang, K. Jönsson, M.A. Kazmi, M.J. Donoghue, T.P. Sakmar, Recreating a functional ancestral archosaur visual pigment, Mol. Biol. Evol. 19 (2002) 1483–1489.

[21] D.J. Parry-Smith, A.W.R. Payne, A.D. Michie, T.K. Attwood, CINEMA—a novel Colour INteractive Editor for Multiple Alignments, Gene 211 (1997) GC45–GC56.

[22] D. Gilis, M. Rooman, PoPMuSiC: an algorithm for predicting protein mutant stability changes. Application to prion proteins, Protein Eng. 13 (2000) 849–856.

[23] K.B. Nicholas, H.B. Nicholas Jr., D.W. Deerfield, GeneDoc: analysis and visualization of genetic variation, EMBNEW News 4 (1997) 14.

[24] C. Damblon, X. Raquet, L.Y. Lian, J. Lamotte-Brasseur, E. Fonze, P. Charlier, G.C. Roberts, J.M. Frere, The catalytic mechanism of beta-lactamases: NMR titration of an active-site lysine residue of the TEM-1 enzyme, Proc. Natl. Acad. Sci. U.S.A. 93 (1996) 1747–1752.