DEMOCRITUS UNIVERSITY OF THRACE
SCHOOL OF HEALTH SCIENCES
DEPARTMENT OF MOLECULAR BIOLOGY & GENETICS

BACHELOR'S THESIS

"A memory-efficient method for quantifying convergence of molecular dynamics trajectories"

Author: Vasiliki Tsampazi, 2575

Supervisor: Dr. Nicholaos M. Glykos
Associate Professor, Laboratory of Structural and Computational Biology, Democritus University of Thrace

Alexandroupolis, Greece, December, 2025

DEMOCRITUS UNIVERSITY OF THRACE
SCHOOL OF HEALTH SCIENCES
DEPARTMENT OF MOLECULAR BIOLOGY & GENETICS

BACHELOR'S THESIS

"A memory-efficient method for quantifying convergence of molecular dynamics trajectories"

Author: Vasiliki Tsampazi, 2575

Supervisor: Dr. Nicholaos M. Glykos
Associate Professor, Laboratory of Structural and Computational Biology, Democritus University of Thrace

I declare that the present thesis entitled "A memory-efficient method for quantifying convergence of molecular dynamics trajectories" is original and was carried out by me personally, as an undergraduate student of the Department of Molecular Biology and Genetics, with Registration Number 2575. I certify that during the preparation and writing of the thesis, all legal requirements were followed, and that the principles of academic ethics and integrity were fully adhered to, which prohibit the falsification of results, the misuse of others' intellectual property, and plagiarism.

Vasiliki Tsampazi

Alexandroupolis, Greece, December, 2025

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΥΓΕΙΑΣ
ΤΜΗΜΑ ΜΟΡΙΑΚΗΣ ΒΙΟΛΟΓΙΑΣ & ΓΕΝΕΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"Ανάπτυξη μιας μεθόδου με αποδοτική χρήση μνήμης για την ποσοτικοποίηση της σύγκλισης τροχιακών μοριακής δυναμικής"

Βασιλική Τσαμπάζη, 2575

Επιβλέπων: Αναπληρωτής Καθηγητής Νικόλαος Μ. Γλυκός
Εργαστήριο Δομικής και Υπολογιστικής Βιολογίας,
Δημοκρίτειο Πανεπιστήμιο Θράκης

Δηλώνω ότι η παρούσα εργασία με τίτλο "Ανάπτυξη μιας μεθόδου με αποδοτική χρήση μνήμης για την ποσοτικοποίηση της σύγκλισης τροχιακών μοριακής δυναμικής" είναι πρωτότυπη και πραγματοποιήθηκε από εμένα προσωπικά, προπτυχιακή φοιτήτρια του Τμήματος Μοριακής Βιολογίας και Γενετικής, με Αρ. Μητρώου 2575. Βεβαιώνω ότι κατά την εκπόνηση της εργασίας και τη συγγραφή της τηρήθηκαν τα προβλεπόμενα από το νόμο, καθώς και ότι ακολουθήθηκαν πλήρως οι αρχές της ακαδημαϊκής ηθικής και δεοντολογίας, οι οποίες απαγορεύουν την παραποίηση των αποτελεσμάτων, την κατάχρηση της διανοητικής ιδιοκτησίας άλλων και τη λογοκλοπή.

Βασιλική Τσαμπάζη

Αλεξανδρούπολη, Ελλάδα, Δεκέμβριος, 2025

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Nicholaos M. Glykos, for introducing me to this fascinating field of Biology and later giving me the opportunity to work in this field under his guidance. His unending patience and mentorship have been invaluable.

I would also like to thank my family and my friends, who may not know exactly what I have been working on, but who have always supported and encouraged me.

Table of contents

Abstract

Molecular Dynamics simulations are a powerful tool when trying to determine the structure of a protein/peptide structure. Many different methods have been introduced for the analysis of the results deriving from Molecular Dynamics simulations, each one based on different principles, with different computational costs. The proposed method that is being presented in this report is based on the "Good-Turing" method created by Koukos and Glykos, but is significantly cheaper computationally. The different methods and tests with which we experimented, as well as the exact details of the "final" method and the quality of its results are presented thoroughly in this report.

Keywords:

– "Good-Turing" method
– convergence
– computational cost

Περίληψη

Οι προσομοιώσεις τροχιακών μοριακής δυναμικής είναι ένα ισχυρό εργαλείο για τον καθορισμό της δομής μιας πρωτεΐνης/δομής ενός πεπτιδίου. Πολλές διαφορετικές μέθοδοι έχουν συσταθεί για την ανάλυση των αποτελεσμάτων που προέρχονται από προσομοιώσεις τροχιακών μοριακών δυναμικών, η κάθε μια βασισμένη σε διαφορετικές αρχές, με διαφορετικό υπολογιστικό κόστος. Η προτεινόμενη μέθοδος, η οποία παρουσιάζεται σε αυτή την αναφορά βασίζεται στην μέθοδο "Good-Turing" των Κούκος και Γλυκός, αλλά είναι σημαντικά πιο φθηνή υπολογιστικά. Οι διαφορετικές μέθοδοι και τεστ με τα οποία πειραματιστήκαμε, καθώς και οι ακριβείς λεπτομέρειες της "τελικής" μεθόδου και η ποιότητα των αποτελεσμάτων της παρουσιάζονται αναλυτικά σε αυτή την αναφορά.

Λέξεις-Κλειδιά:
– μέθοδος "Good-Turing"
– σύγκλιση (convergence)
– υπολογιστικό κόστος

1.Introduction

1.1 Molecular Dynamics Simulations

The experimental structure determination methods (X-ray crystallography, cryo-EM, NMR) provide excellent information on a specific structure of a protein (like taking a still of a movie scene), but, just like every molecule, proteins are dynamic and their structure changes depending on the environment they are surrounded by. MD (molecular dynamics) simulations are based on the physics pertaining to the interatomic interactions and that is why they are a very powerful tool when it comes to the study of proteins. With MD simulations it is feasible to predict how a protein behaves in different solutions, with or without the presence of various ligands and other molecules that interact with a protein's surface and how a protein folds and/or unfolds. That, of course, does not mean that the experimental structure determination methods are not useful. Data from experimental structure determination methods are being used for the MD simulations and the quality of these data plays a crucial role in the overall quality of the MD simulation's results. Also, results of MD simulations can lead to new experiments with experimental structure determination methods in order to further investigate a structure that was first depicted in a MD simulation[1]. One important parameter that affects the quality and the accuracy of the MD simulations' results is the simulation time (i.e. for how many μs, ps or ns the MD simulations run). As one can very easily deduce, the longer the simulation time, the better, since even rare conformations will appear and will be taken into consideration as they should. The issue with the increase of simulation time is the fact that it is extremely expensive computationally. Since there is no absolute and collective rule when it comes to the simulation time, how can one know if all important molecular configurations have been observed? And even if there is a reliable method for determining convergence, is there a way it can also be computationally cheap? Based on the work by Koukos & Glykos[2], there has been an attempt to answer these questions.

1.2 The classic "Good-Turing" method

What Koukos & Glykos proposed is a "probabilistic measure of convergence"[2]. Their program analyses trajectories of proteins and/or peptide

structures using RMSD matrices to measure distance between the different conformations. The sample is analysed so the maximal RMSD (meaning the RMSD value that corresponds with the biggest conformational difference observed in the sample at hand) is determined for different sampling factors. Through weighted nonlinear least-squares fitting, the optimal sampling factor is calculated and its submatrices are used for the further evaluation of the sample, which consists of a hierarchical clustering method and the Good-Turing statistics[11]. Thanks to the dendrogram, clusters of the observed formulations are being produced easily and with the application of the Good-Turing formalism the probability of unobserved conformations for different RMSD values is calculated[2], ultimately resulting in the desired probability of unobserved species ($P_{unobserved}$) vs RMSD distribution. The referred "submatrices" are the submatrices that occur based on the sampling factor that has been determined as the optimal one. At this point, it should be highlighted that if the program cannot produce an optimal sampling factor, the sample is considered insufficient and the analysis stops there. One great feature of this method is the fact that not only does it answer whether convergence has been achieved[2,3] or not, but it also calculates how different the conformations that will be observed be from those already observed, if someone were to double the simulation time. That way, one can decide whether it is worth the computational cost to double the simulation time or if the quality of the results is good. Simulation time is not the only parameter to ensure sufficient sampling; using statistically independent observations is critical for good sampling quality and for an overall good set of data[3]. In every trajectory used for a MD simulation, some conformations are correlated with each other and thus should not be analysed as separate formulations. One way of dealing with the issue is by grouping the correlated conformations together, that way ending up with a number of independent groups which can be properly analysed. In the program that Koukos & Glykos[2] made, it is possible for the user to decide the step (stride) between frames–which naturally affects the matrix size–and with the sub-sampling factor that the program "determines" itself, a specific–or put a little differently–an actual step (stride) is being set to read the RMSD matrix depending on the step (stride) that the user opts. But, as robust as the Good-Turing program is, it is also quite expensive computationally. Research has been conducted so an alternative method can be introduced that still delivers trustworthy results and is also computationally cheaper than the classic "Good-Turing" method. Of

course, it is crucial to point out that any computationally cheaper method, although satisfactory, will always require a sacrifice in accuracy.

1.3 The “max of mins” method

In an attempt to find the method that would satisfy the requirements stated above, the “max of mins” method was first proposed. For the “max of mins” method, the same program–grcarma–is used, but in this case, out of every line of the RMSD submatrices constructed (based on the optimal sampling factor) only the maximum RMSD value observed is a branch of its own in the dendrogram that will follow. That way, the computational cost can be significantly reduced, but there is a sacrifice in accuracy. How big is that sacrifice? The first step to test the potential of the method, that has just been proposed above, is to use the already existing Good-Turing program with the parameters set in such a way that two different sets of data occur from each run; one set of data would derive from the classic “Good-Turing” method and the other would derive from the “max of mins” method. The comparison of the two datasets (two datasets from each run), for the different matrix sizes of the same protein tested each time, provided information regarding the “max of mins” method, as well as the accuracy of the classic “Good-Turing” method. The purpose all these “different matrix sizes of the same protein tested each time” serve, is that they provide information regarding the accuracy and the agreement of the results between different sized datasets for both methods and they also serve as an indication of the ideal step (stride) between frames (and consequently matrix size), so the final results are satisfactory and the computational cost is as low as possible. The sampling and the results obtained from both of those methods are discussed further in the sections below.

1.4 The “independent” method

At this point, it is important to highlight that, as it has been stated, the results obtained from the “max of mins” method are not independent from the classic “Good-Turing” method. So, the “max of mins” method can work very well as an indicator of the potential of the “max of mins” logic, as a computationally cheaper but still reliable method, but the “max of mins” method itself is not

computationally cheaper. An actually computationally cheaper method based on the “max of mins” logic and at the same time independent of the classic “Good-Turing” method, is a method that for the purposes of this report will be called the “independent” method from now on. The “independent” method is a program written by Glykos in the command line (of a Unix cell) and the results obtained are being produced thanks to the carma program. The most important parameters that are responsible for the quality and the nature of the results deriving from this method (the “independent” method) are the step (stride) between frames selected by the user and the use of the maximum RMSD value observed out of every line of the matrix as a branch of its own, in the dendrogram that follows in order to reduce the computational cost. It is also crucial to note that the first structure opted as the reference structure is being automatically changed due to the step (stride) set by the user via the “$i” variable of the code.

1.5 The “Jude” method

After the results of the “independent” method, for different proteins, run with various steps (strides) between frames, enough data had occurred so a more complete, refined and detailed method was implemented, the “Jude” method. The “Jude” method is a program written in perl and just like the “independent” method’s code, it was written by Glykos. The results of the “Jude” method were analysed and the datasets regarding the probability of unobserved conformations for different RMSD values ($P_{unobserved}$ vs RMSD distribution) were compared with the results that derived from the respective datasets produced with the classic “Good-Turing” method. In contrast to the rest of the methods that were tested for the purposes of this research, the “Jude” method has an extra parameter, the a_3 parameter, which works as an indication of convergence. The a_3 parameter is one of the parameters of the equation used to perform the weighted nonlinear least-squares fitting of the data. After the fitting of the data has been completed, what follows is the segmentation of the line, which happens in order to indicate where the plateau is. The largest line out of the rest ideally indicates the plateau and as long as this is true, the actual step (stride) between frames is determined and then the rest of the program’s procedure is practically the same with the one in the “max of mins” method.

2. Methods

2.1 The classic “Good-Turing” method and the “max of mins” method

With the grcarma[44] program, runs for different proteins at different matrix sizes each time were executed and two sets of data were produced, one for the classic “Good-Turing” method and another for the “max of mins” method. The matrix size varied from a range of approximately 10000 x 10000 up to 30000 x 30000 and the results that occurred from these runs consisted of data files for the classic “Good-Turing” method, as well as for the “max of mins” method. From these grcarma runs, the evaluation of two different parameters was possible; the quality of the results in the different matrix sizes–as it has been mentioned before, the size is determined based on the step (stride) between frames that the user sets–and the quality of the results produced in the two methods (classic “Good-Turing” and “max of mins” method). It is important to evaluate whether the different matrix sizes affect the quality of the results in any way that matters, because the classic “Good-Turing” method works as a reference point whilst trying to find this sought out computationally cheaper, but also robust method. With the optimal sampling factor already determined (thanks to the run of the classic “Good-Turing” method) the “max of mins” method utilizes the respective submatrices but only the maximum RMSD value of each is being considered when constructing the dendrogram and implementing the Good-Turing formalism. The analysis of these maximum RMSD values, with the Good-Turing formalism, produces the desired $P_{unobserved}$ vs RMSD distribution (i.e. the probability of unobserved conformations for different RMSD values).

2.2 The “independent” method

The first actually computationally cheaper method tested was the “independent” method. The “independent” method is based on the “max of mins” logic, meaning that out of every submatrice only the maximum RMSD value of each is used for the further analysis of the sample with the carma[45] program (and subsequently the Good-Turing formalism). The essential difference between the “max of mins” method and the “independent” method is that in the latter, the

optimal sampling factor is not known beforehand and there is no sure way of knowing that the sampling factor used for the analysis is the optimal one, since no analysis is being conducted to determine it. The only parameter that determines which sampling factor (remember not necessarily the optimal sampling factor) is going to be used for the analysis of the trajectory, is the step (stride) between frames set by the user. In order to test this method different steps (strides) between frames were selected which ranged from 10000 to 300000. The "independent" method was written in such a way that it is meant to be executed in the linux command line and the user can adjust the code directly.

```
for i in (seq 1 500 49000)
     carma -v -fit -cross -mm -first $i -step 50000
file_name.dcd file_name.psf
     sort -n carma.RMSD.mins | awk -v noflines=(wc -l <
     carma.RMSD.mins) '{print $1,1.0-(NR/noflines)}' >>
     out.dat
     end ; rm carma.RMSD.mins density.log.matrix
     density.matrix
```

The step (stride) between frames chosen not only reduces the computational cost, as the whole trajectory will not be analyzed, but it also prevents correlated conformations from being analyzed as independent ones. As it has been stated above in this section, the "independent" method is based on the "max of mins" method, so the carma program is being used for the clustering of the maximum RMSD values of the submatrices the same way it occured in the "max of mins" method and then the Good-Turing formulation is being utilized for the final analysis of the sample, which produces the desired $P_{unobserved}$ vs RMSD distribution (i.e. the probability of unobserved conformations for different RMSD values). The datasets produced by the "independent" method were compared with the ones from the classic "Good-Turing" method, in order to evaluate the performance of the former.

2.3 The “Jude” method

Based on the results deriving from the “max of mins” and the “independent” method–which are being discussed in the corresponding sections–, the “Jude” method has a fixed step (stride) between frames, so immediately the computational value decreases significantly, since the maxRMSD vs sampling factor distribution is not being calculated. Consequently, with the sampling factor fixed, what follows is the analysis of the submatrices in order to find the maximal RMSD for each superdiagonal and the construction of the maximum RMSD vs superdiagonals distribution. Then, the fitting of the data from the maximum RMSD vs superdiagonals distribution is being executed with the help of a weighted nonlinear least-squares equation, which is based on the weighted nonlinear least-squares equation used in the classic “Good-Turing” method[2], but for this method, it has been modified in such a way by Glykos, that an extra parameter is being calculated in this method, called a_3. The a_3 parameter works as an indicator of convergence, meaning that it indicates whether the simulation time is good, or if it needs to be longer. More specifically, the simulation time is good if the value of the a_3 parameter is greater than 0.98 (>0.98). So long as the value of the a_3 parameter is greater than 0.98, the analysis continues smoothly, where the line of the maximum RMSD vs superdiagonals distribution is segmented in a number of lines. The largest segment that occurs in the distribution, represents the plateau. It is possible that the largest segment’s ends of the distribution indicate both the beginning and the end of the plateau, but if at least one of the ends indicates the beginning or the end of the plateau, depending on the slope of said segment, the method is satisfactory and works just as it is intended to. With either of the plateau’s limits determined, it is possible to calculate the actual step (stride) between frames that needs to be set for the following analysis of the sample. This actual step (stride) between frames is determined by the following formula: the segment’s end’s value which equates to the plateau x 800 (the initial step set by the program for the analysis of the data thus far). The final results of this method that answer the same question that the classic “Good-Turing”, the “max of mins” and the “independent” method all did–i.e. the probability that a conformation important for the analysis in hand has not yet been observed– are produced thanks to the analysis of the sample– with the “max of mins” logic (the way it works has been discussed above), only now the actual step (stride) between frames is not determined through a maximum RMSD vs sampling factor distribution which

was significantly computationally expensive (classic “Good-Turing” method, “max of mins” method), nor through random trials that were unreliable (“indepented” method). The results produced by the “Jude” method have been compared with the results deriving from the classic “Good-Turing” and the conclusions pertaining the method’s efficacy are discussed below in the respective sections.

a3<0.98	simulation time needs to be longer	convergence has not been reached and/or is questionable
a3=0.98	simulation time is adequate	convergence has been reached
a3>0.98	simulation is time adequate	convergence has been reached

Table 1. The a3 parameter as an indication of convergence.

The final results of the “Jude” method that include a $P_{unobserved}$ vs RMSD distribution (probability of unobserved conformations for different RMSD values), are produced thanks to a final analysis of the sample with the “max of mins” logic, where the actual step, determined by the previous analyses that took place (in this method), is used as the step that will be used for that part of the analysis. So, the starting step (stride) between frames might be “fixed” at 800 and the superdiagonals at 150, but the actual step (stride) between frames used for the production of the final results is not the exact same for each peptide structure. The program written for the “Jude” method can be found in the github repository at https://github.com/glykos/GoodTuring.

2.4 The nature of the "Jude" method's samples

A brief overview for the peptide structures presented follows in order to further explain the results:

- cln-ILDN (PDB: 2RVD) is a stable, linear, 10-peptide protein[22,33]. cln-ILDN is also a synthetic construct, and the purpose for its construction was to be the most stable peptide known, whose structure would be determined by experimental methods.
- A31P (PDB:1B6Q, 1GMG)[24,26,41] is a mutated version of the native protein Rop (Repressor of primer), where Ala31 has been replaced by a Pro residue. Rop is found in *Escherichia Coli*, and it is a 63-amino acid protein. Both Rop and A31P are usually found bound together with another monomer of the same protein (Rop and A31P respectively), but in contrast to Rop, A31P is not stable. Data from structure determination methods show that A31P does not reach convergence and is almost constantly unfolded. The Ala31 to Pro mutation leads to a right-handed, mixed parallel and antiparallel bundle, displaying a "bisecting U" topology, whereas the native Rop is a left-handed, all-antiparallel 4-α-helical bundle. At this point, it is also important to mention that although stable, the Rop protein folds slowly, which is part of the reason why it was initially difficult to determine whether the A31P mutant folds slowly as well, forming a similar structure to the Rop one, or if it is mostly unorganized and unfolded. The latter turned out to be true.
- 2n0x (PDB:2N0X) is a structurally flexible, 16-amino acid fragment of serum albumin 2n0x, generated when found in acidic environments of the organism (humans/homo sapiens) and works as an inhibitor of CXCR4[23]. The 2n0x protein is not considered a stable peptide structure and it takes a significant amount of time for it to reach convergence.
- 6NM2[25,35] (PDB:6NM2) is a small (9-residue) synthetic, antimicrobial protein. A two-turn helix and a Trp triplet (WWW) that forms a π configuration are the most noteworthy parts of the peptide. The hydrophobic parts of the Trp triplet (WWW) are the ones found permeating the membranes and the aromatic-aromatic, as well as, the aromatic-aliphatic interactions in the triplet, play an important role in its stability. Overall, the amphipathic 6NM2 peptide can be considered stable

and its antimicrobial activity especially against *Staphylococcus aureus* is mostly thanks to the aforementioned Trp triplet (WWW).

- NAT peptide structures: Enzyme structures that possess a NAT (N-terminal acetyltranferases) terminal have been tested (Nat-STAR-notails, Nat-STAR, NatA31P)[30]. The NAT structure gives the N-terminal a neutral charge, a hydrogen bond acceptor, it affects the α-amino nitrogen nucleophilicity and basicity and is also responsible for an increase in size and hydrophobic nature. Every NAT terminal has a peptide binding site and the hydrogen bonds observed in the first 2-3 N-terminal residues offer more stability to the structure.
- pdb2mq2 (PDB: 2MQ2) is an antimicrobial protein who targets the membranes of both Gram-negative and Gram-positive bacteria[27,36]. 2MQ2 or also known as CDP-1 (Cysteine Deleted Protegrin-1) is a mutated version of Protegrin-1 (PG-1), where the cysteines 6, 8, 13, 15 of PG-1 are deleted and consequently the disulfide bonds between them (Cys6-Cys15 & Cys8-Cys13), which provided stability to the rigid β-hairpin like structure. The deletion of Cysteines did not eliminate the antimicrobial activity, because although the deletion caused conformational changes, the β-hairpin like structure can still be observed. Of course, the β-hairpin-like structure of the PG-1 is not identical to the one found in CDP-1; in CDP-1 there is a chain reversal in the β-hairpin structure, some sidechains look as if they are outwards of the β-hairpin structure and interactions that are most likely hydrogen bonds exist/are formed.
- NFGAILS (PDB:5E5V) is part of the Islet Amyloid Polypeptide (IAPP) and more specifically it is the segment of the protein that consists of residues 20-29 and is in the amyloidogenic C-terminal region[29,37]. In this region the peptide structure consists of anti-parallel β-stands arranged into parallel β-sheets, which form a class 7 out-of -register steric zipper. The hydrogen bonds in this zipper are responsible for the strong and weak interface observed, which does not offer much stability in this area.
- SarsTM (PDB:7K3G) is the transmembrane (TM) domain of Sars-CoV-2's envelope protein E. Protein E forms a homopentameric cation channel[31,32,38]. In particular, the three Phe (Phe20, Phe23 and Phe26), the Val17 and the Leu31 residues of the structure's center are responsible for the stability and hydrophobic nature of the structure, but overall the E protein cannot be considered as a particularly stable protein.

The α-helical core in the TM domain, alongside with the hydrophobic residues mentioned above of this 30-residue TM domain, are the key elements that make it a fast, stable helix forming domain.

- InflA (PDB: 4QK7, 2H95, 2L0J**)** is a transmembrane peptide of the Influenza A virus[32]. This structure is very flexible with its TM domain being the most flexible area of it (residues 22-46). Both the N-terminal and the C-terminal are highly flexible and the homotetrameric proton channel pores form an α- helix, which again is very flexible, meaning that many different α-helical conformations are observed when running MD simulations. The only reason why the C-terminal and the N-terminal have a slightly different flexibility is because of a Gly residue that disrupts the α-helix. The viroporin described, is an interesting peptide structure, since it allows for the attack of the host through assistance in the unpacking of the viral genome.
- 5glh (PDB: 5GLH) or else known as Endothelin-1 (ET-1) is an endogenous agonist and binds to the ET_A and ET_B receptors, which belong in class A G-protein-coupled receptors (GPCRs) family[28,40]. The protein consists of hydrophobic amino acids (Ile, Trp) and polar amino acids (Asn, Gln, Asp) which reinforce the protein's strong binding to its receptor (ET_B) through the hydrophobic interactions and the hydrogen-bonds that occur. The electrostatic interaction network at the C-terminal, is what makes the protein unstable when "free", but further stabilises the complex (ligand and receptor). Another very important feature of the C-terminal is the CWXP motif, which plays a crucial role in the protein's ability to work as a ligand (without the Trp21 amino acid the protein cannot work as a ligand). All in all, the protein is not extremely stable (the N-terminal and the α-helical region of ET-1 are the regions that can be characterized as stable) and the amino acids that contribute to its flexibility are essential for its proper function (the C-terminal region is the particularly flexible region).

3. Results

3.1 The classic “Good-Turing” method and the “max of mins” method

For the “max of mins” and the “independent” method a plethora of different proteins (as samples) was not necessary, in order to evaluate the efficiency of the methods, since the “max of mins” method was not actually computationally cheaper and the “independent” method’s results were such, that indicated the need for another method. Therefore, trajectories for two different peptide structures, one significantly stable (cln025) and one significantly flexible (2n0x), were sampled for these two methods. Another thing that should be highlighted once more, is the fact that when the “max of mins” method was tested, the classic “Good-Turing” method was tested as well, in terms of the different matrix sizes and the quality of the results depending on them. For both the cln025[46] and the 2n0x peptide structures, different steps (strides) between frames were opted, which resulted in different sized matrices and different optimal (sub-sampling) factors.

cln025:

The strides (steps) between frames that have been opted and the corresponding matrix sizes are the following:

Stride (step) between frames	Matrix size
610	10345x10345 matrix
435	14507x14507 matrix
420	15025x15025 matrix
320	19720x19720 matrix
310	20356x20356 matrix
250	25241x25241 matrix
210	30050x30050 matrix

Table 2. Step (stride) between frames & matrix size of the samples for the cln025 peptide structure.

The optimal (sub-sampling) factors of the samples do not consistently decrease as the matrix size gets smaller as it is evident in Table 3.:

Sub-sampling factor of:	Matrix size
10	10345x10345 matrix
15	14507x14507 matrix
13	15025x15025 matrix
19	19720x19720 matrix
17	20356x20356 matrix
18	25241x25241 matrix
24	30050x30050 matrix

Table 3. Optimal (sub-sampling) factor & matrix size of the samples for the cln025 peptide structure.

The distributions that occurred out of the datasets obtained from the different runs illustrate how robust the “max of mins” method is and how reliable smaller matrices are.

When it comes to the data deriving from the “max of mins” method for the cln025 peptide structure the distributions are the following:

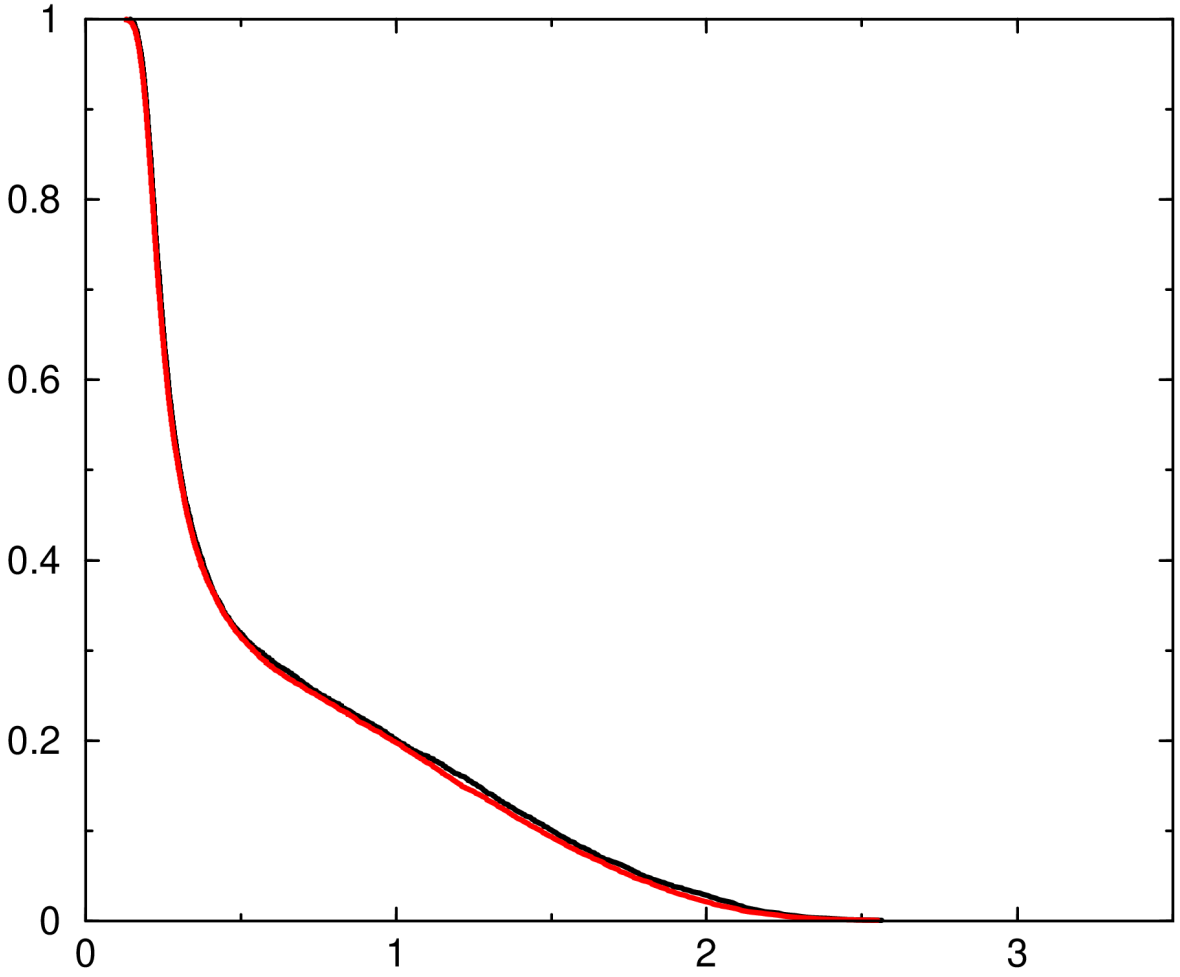

Fig.1 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 30050x30050 matrix (red line) for the cln025 peptide structure.

The lines are very close with each other and in some areas of the graph they overlap.

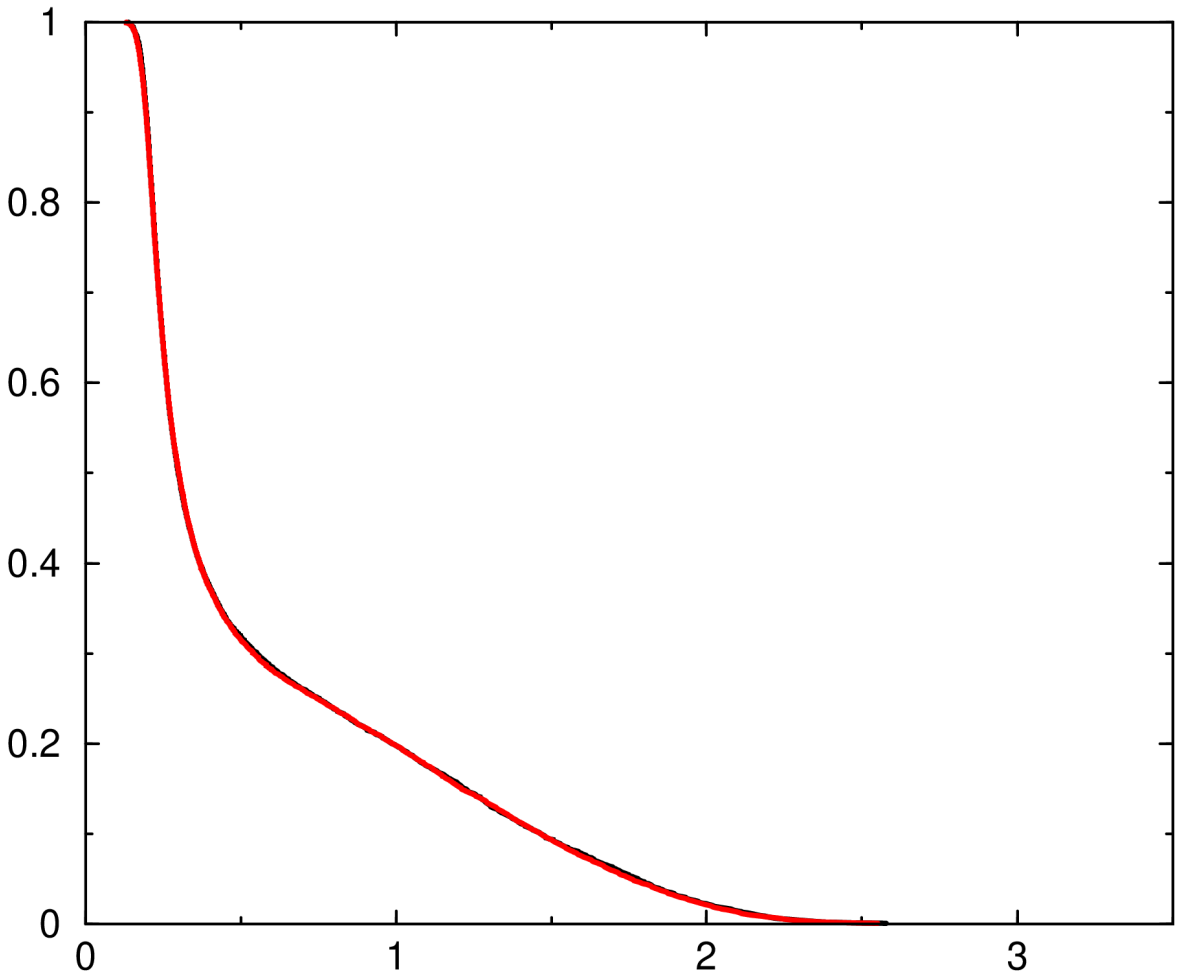

Fig.2 Comparison between the results obtained using a 15025x15025 matrix (black line) and a 30050x30050 matrix (red line) for the cln025 peptide structure.

The lines overlap almost perfectly. It should also be noted, that the lines of Fig.1 are remarkably close but the lines in Fig.2 overlap better.

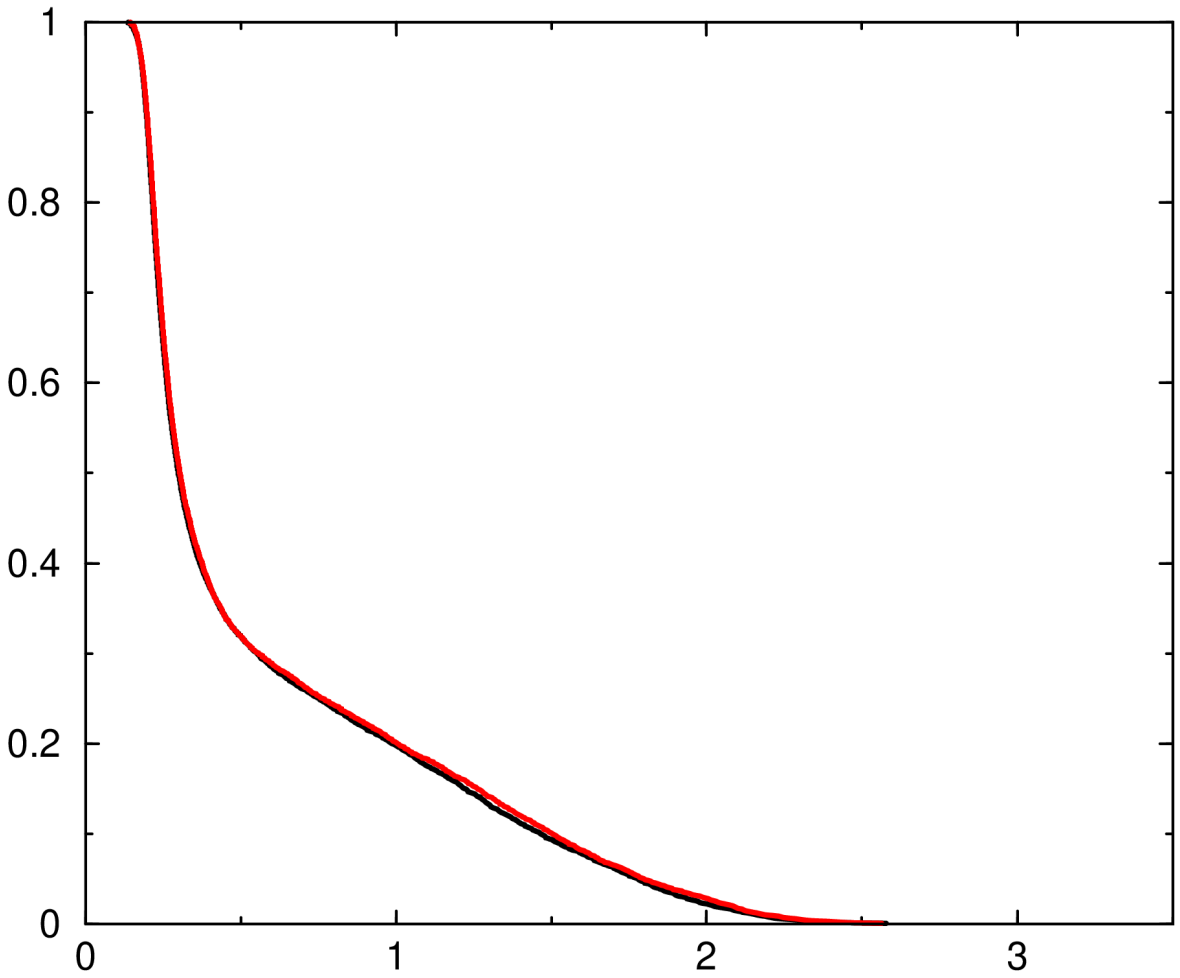

Fig.3 Comparison between the results obtained using a 15025x15025 matrix (black line) and a 10345x10345 matrix (red line) for the cln025 peptide structure.

The lines are very close and they overlap in almost every area of the graph, but the lines in Fig.2 overlap better.

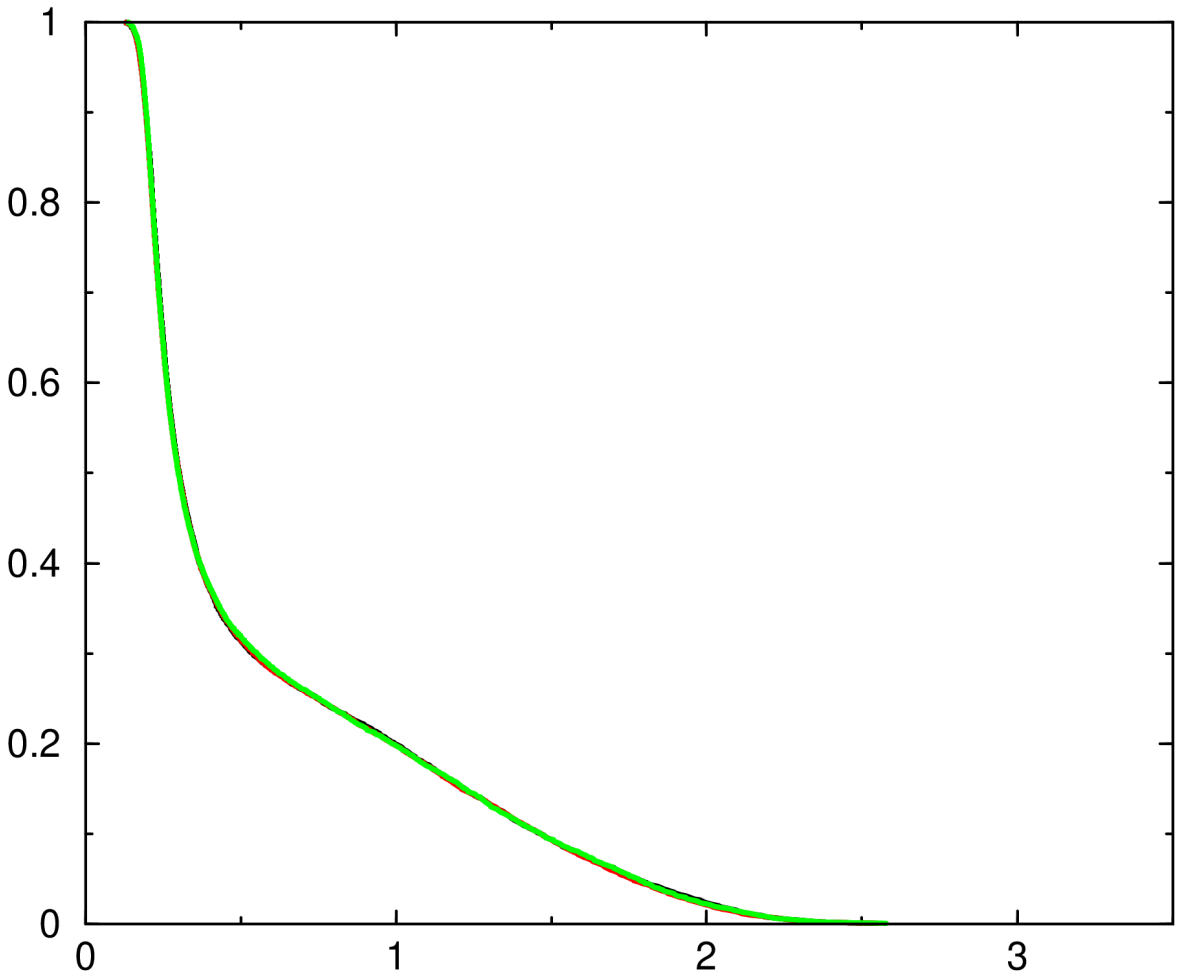

Fig.4 Comparison between the results obtained using a 20356x20356 matrix (black line), a 30050x30050 matrix (red line) and a 15025x15025 matrix (green line) for the cln025 peptide structure.

The lines overlap almost perfectly.

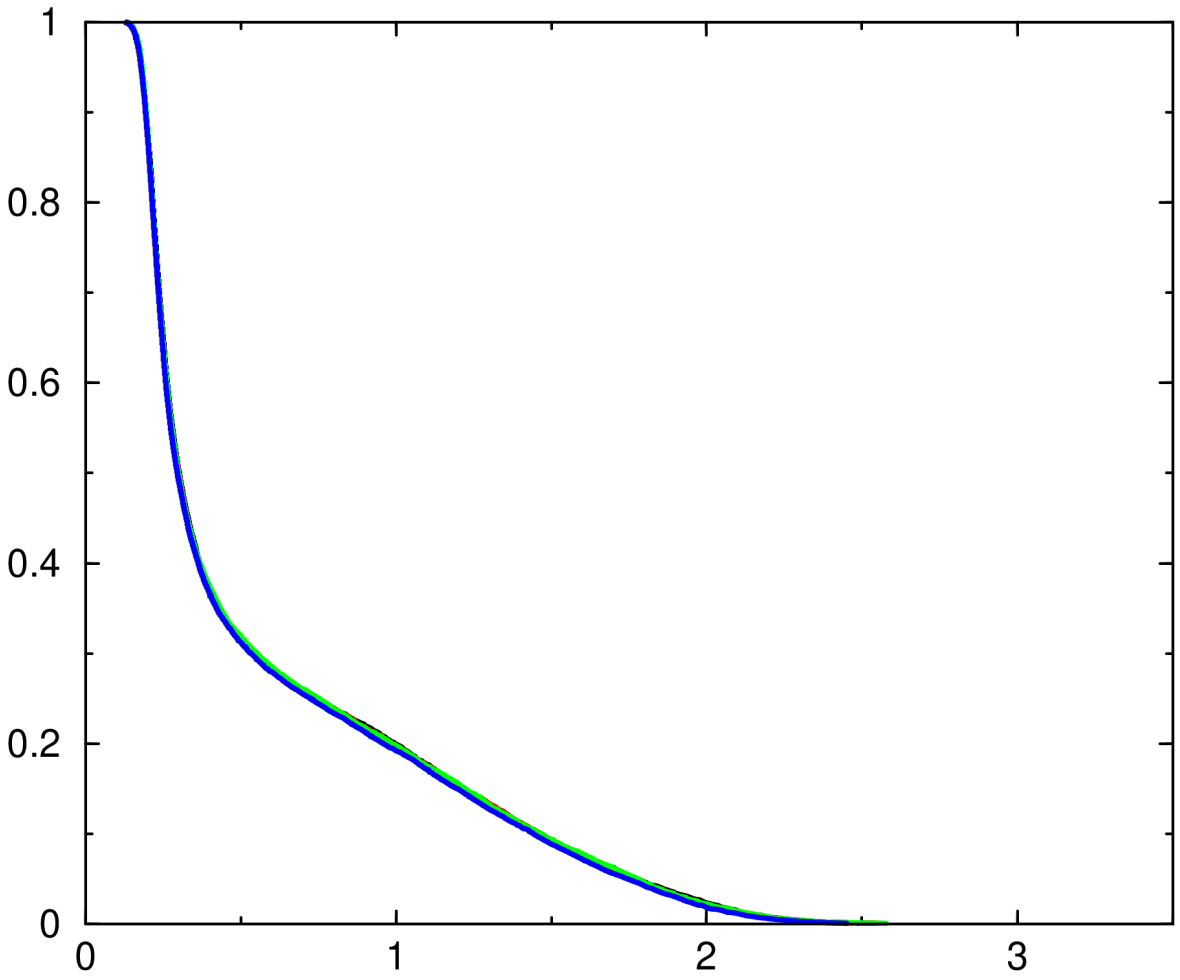

Fig.5 Comparison between the results obtained using a 20356x20356 matrix (black line), a 30050x30050 matrix (red line), a 15025x15025 matrix (green line) and a 25241x25241 matrix (blue line) for the cln025 peptide structure.

The blue line is slightly lower (has a slightly lower set of values) than the other lines.

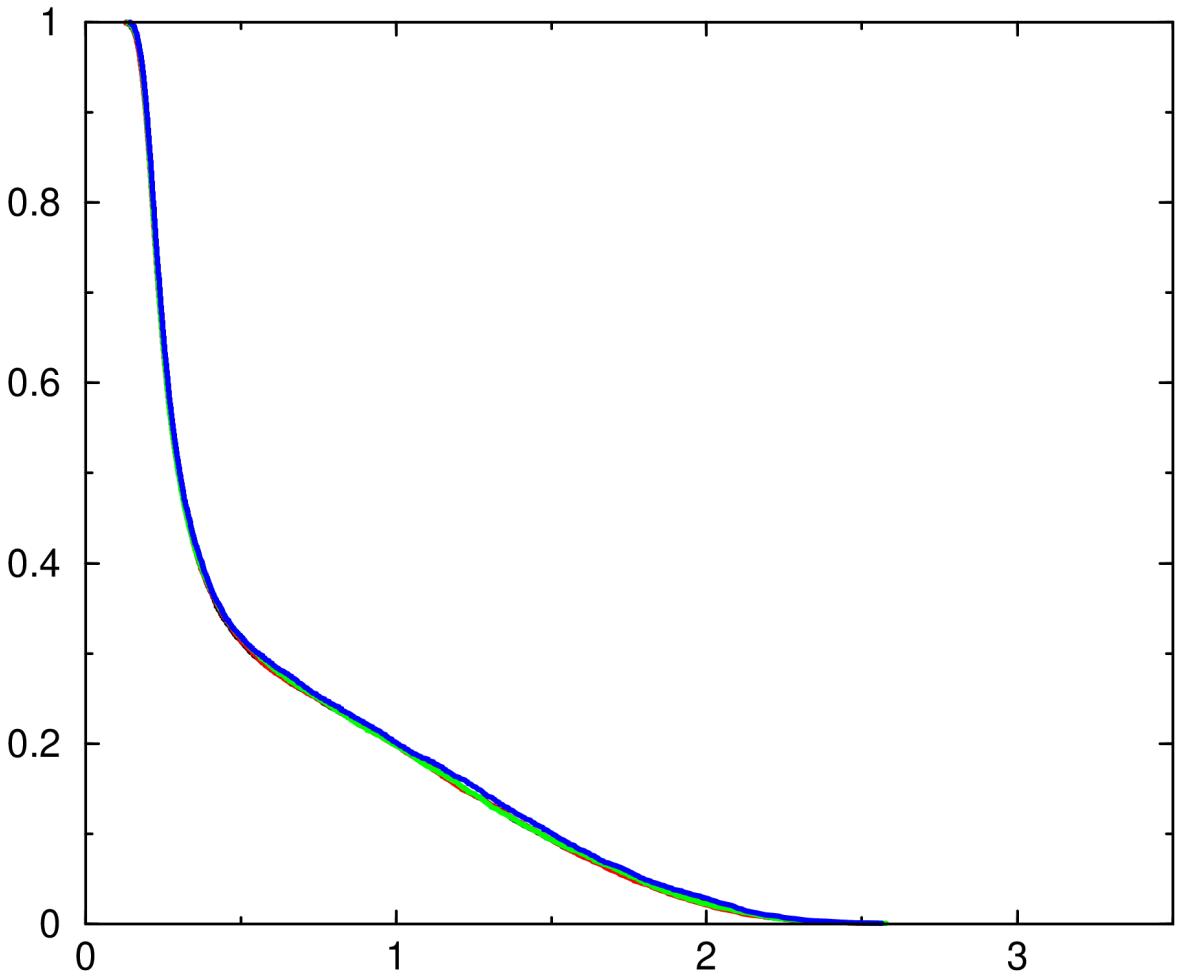

Fig.6 Comparison between the results obtained using a 20356x20356 matrix (black line), a 30050x30050 matrix (red line), a 15025x15025 matrix (green line) and a 10345x10345 matrix (blue line) for the cln025 peptide structure.

The blue line is a little higher (has a set of higher values) than the other lines of the graph, which seem to overlap almost perfectly. The distance between the blue line of Fig.5 and the other three lines of the graph is smaller than the one between the blue line of this graph (Fig.6) and the other three lines. The “other three lines” derive from the exact same dataset, which is why the comparison is being made.

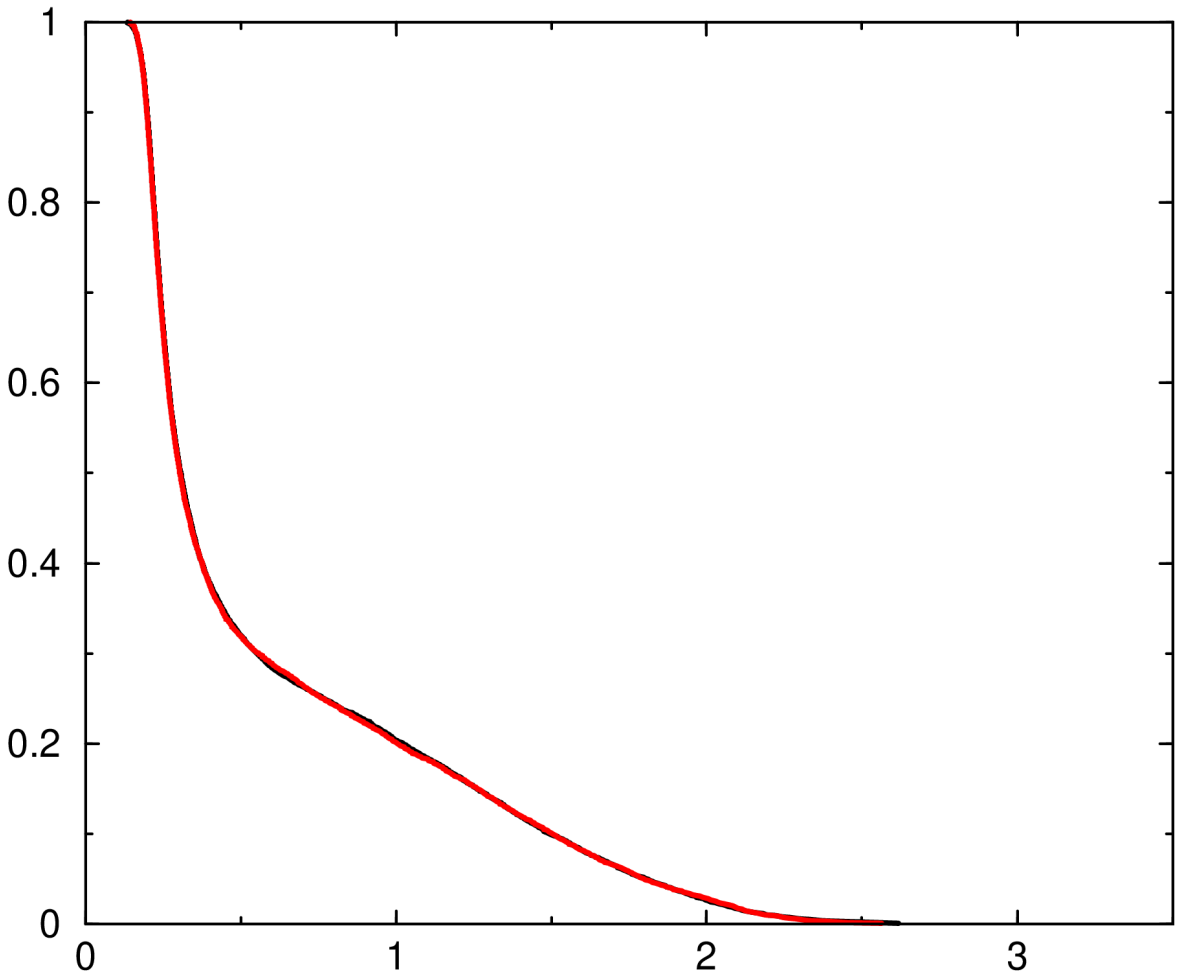

Fig.7 Comparison between the results obtained using a 14507x14507 matrix (black line) and a 10345x10345 matrix (red line) for the cln025 peptide structure.

The lines overlap almost perfectly. The lines that derive from the datasets of the 15025x15025 and 30050x30050 matrices (Fig.2) and the lines that derive from the datasets of the 14507x14507 and 10345x10345 matrices (Fig.7) overlap better than any other pair.

The graphs deriving from the classic “Good-Turing” method for the cln025 peptide structure are the following:

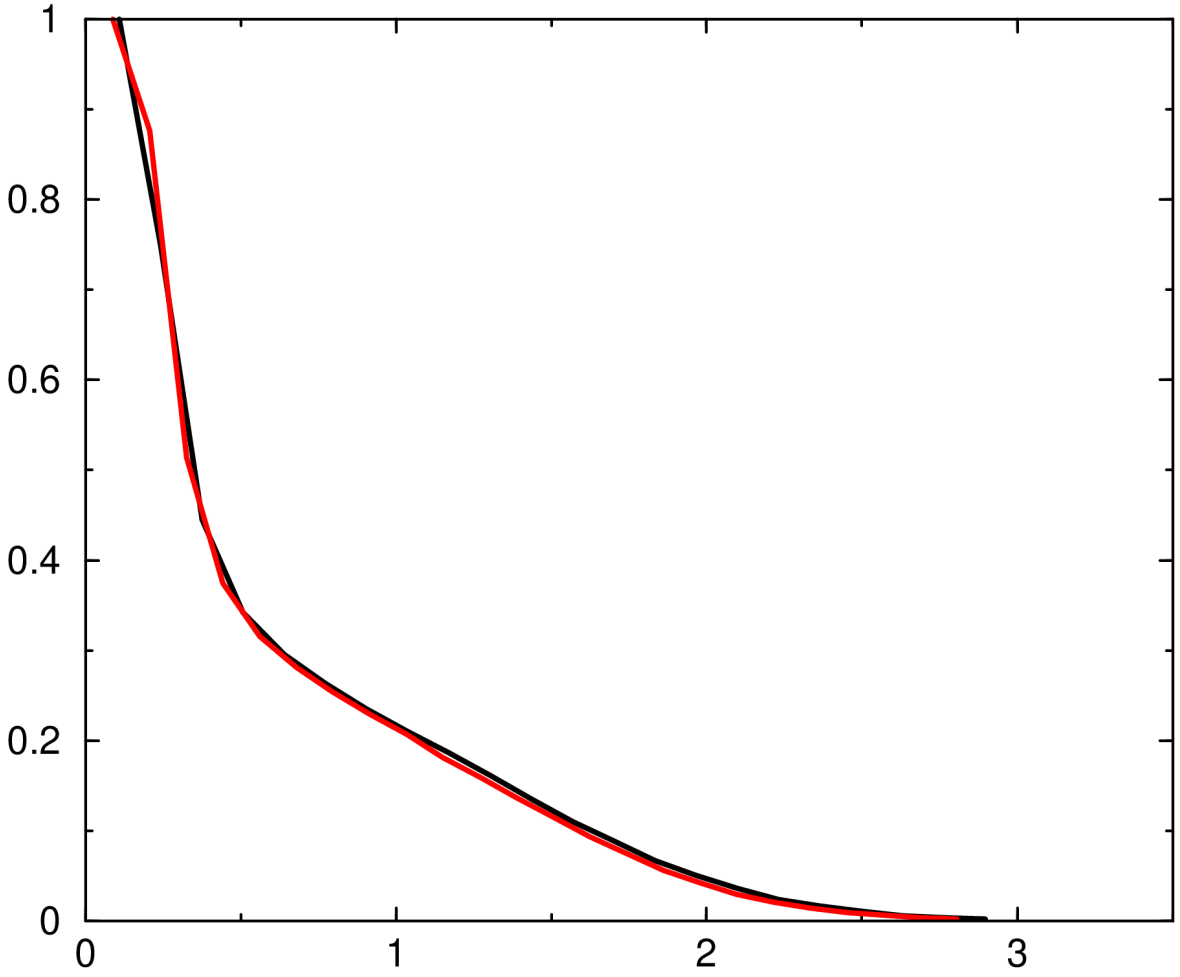

Fig.8 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 30050x30050 matrix (red line) for the cln025 peptide structure.

The lines are very close to each other and there is a slight overlap in various sections of the graph.

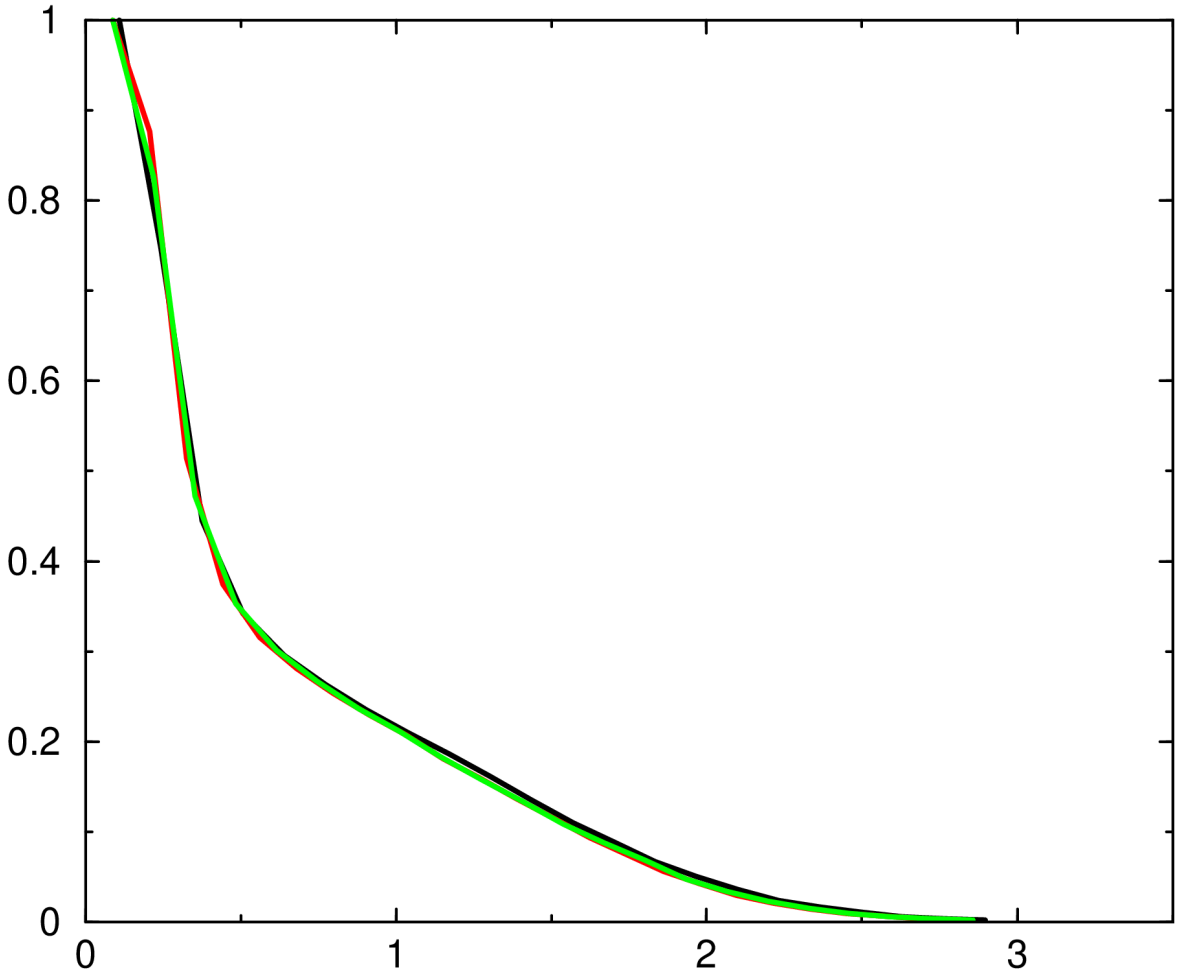

Fig.9 Comparison between the results obtained using a 10345x10345 matrix (black line), a 30050x30050 matrix (red line) and a 15025x15025 matrix (green line) for the cln025 peptide structure.

The green line looks almost between the red and the black lines. More specifically, the green line appears to be a little closer to the black line at the lower RMSD values and a little closer to the red line at the higher RMSD values. The overlapping of the lines is obvious in this figure (Fig.9) as well.

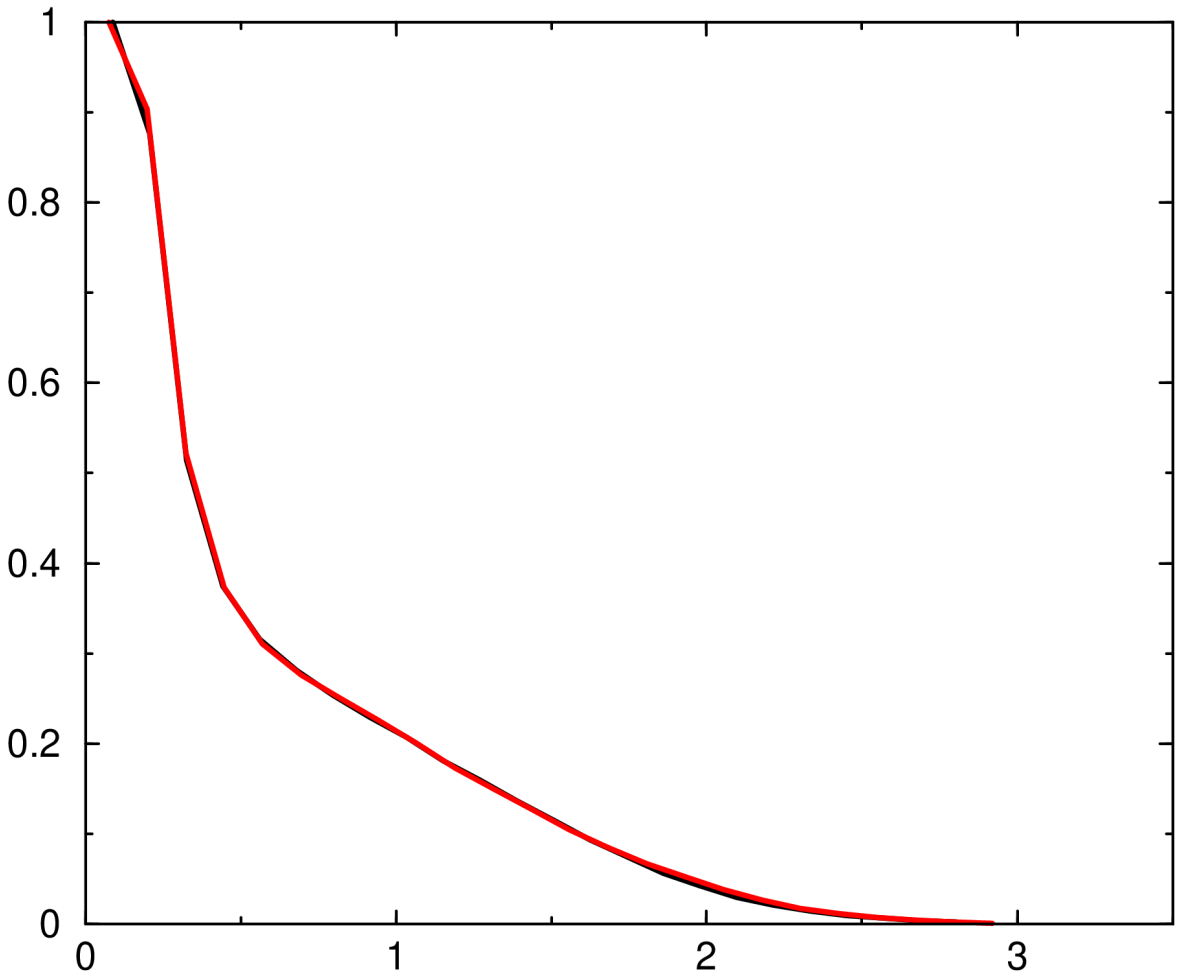

Fig.10 Comparison between the results obtained using a 30050x30050 matrix (black line) and a 20356x20356 matrix (red line) for the cln025 peptide structure.

Apart from an area at the lower RMSD values, the lines overlap.

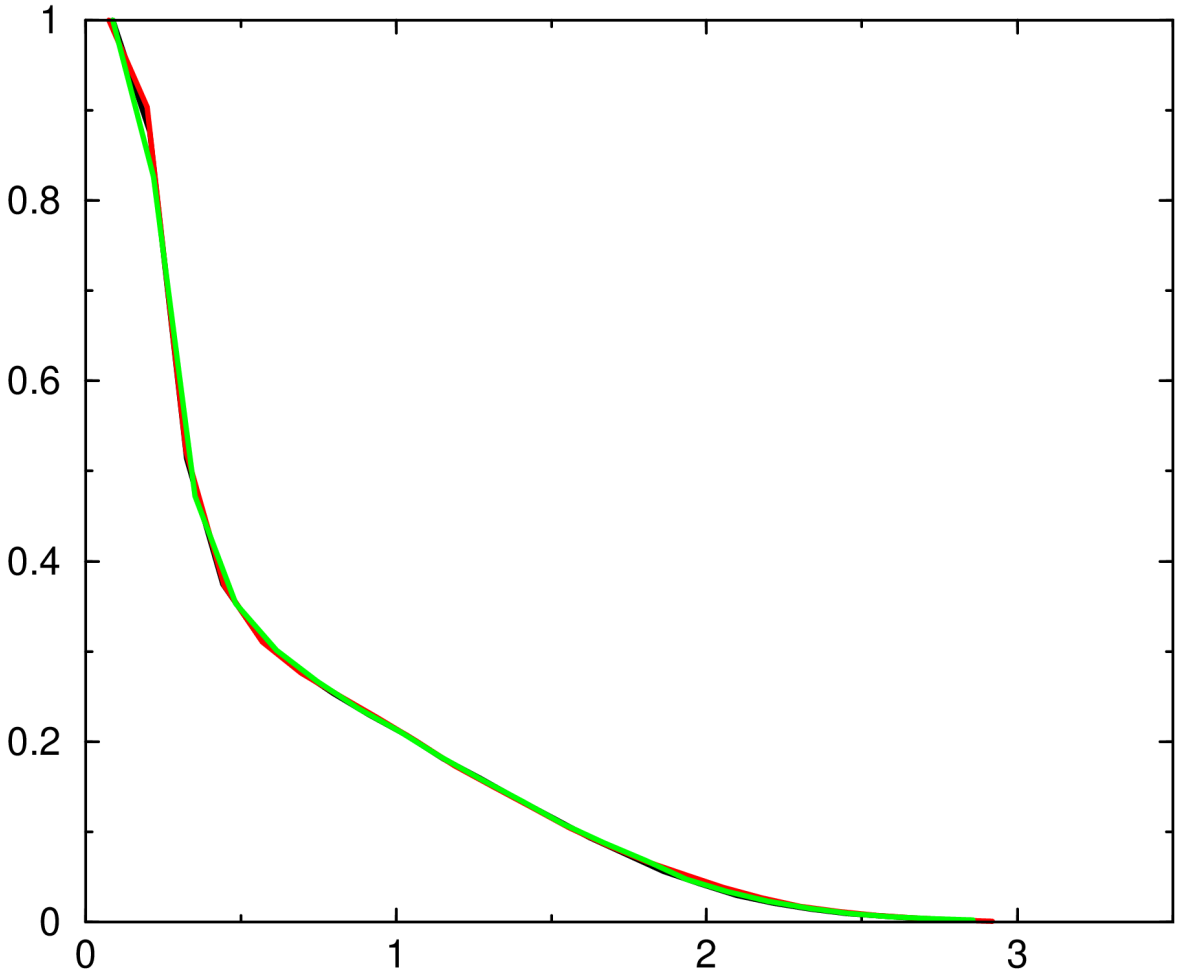

Fig.11 Comparison between the results obtained using a 30050x30050 matrix (black line), a 20356x20356 matrix (red line) and a 15025x15025 matrix (green line) for the cln025 peptide structure.

The lines overlap in many areas and in the ones where they do not, they are remarkably close.

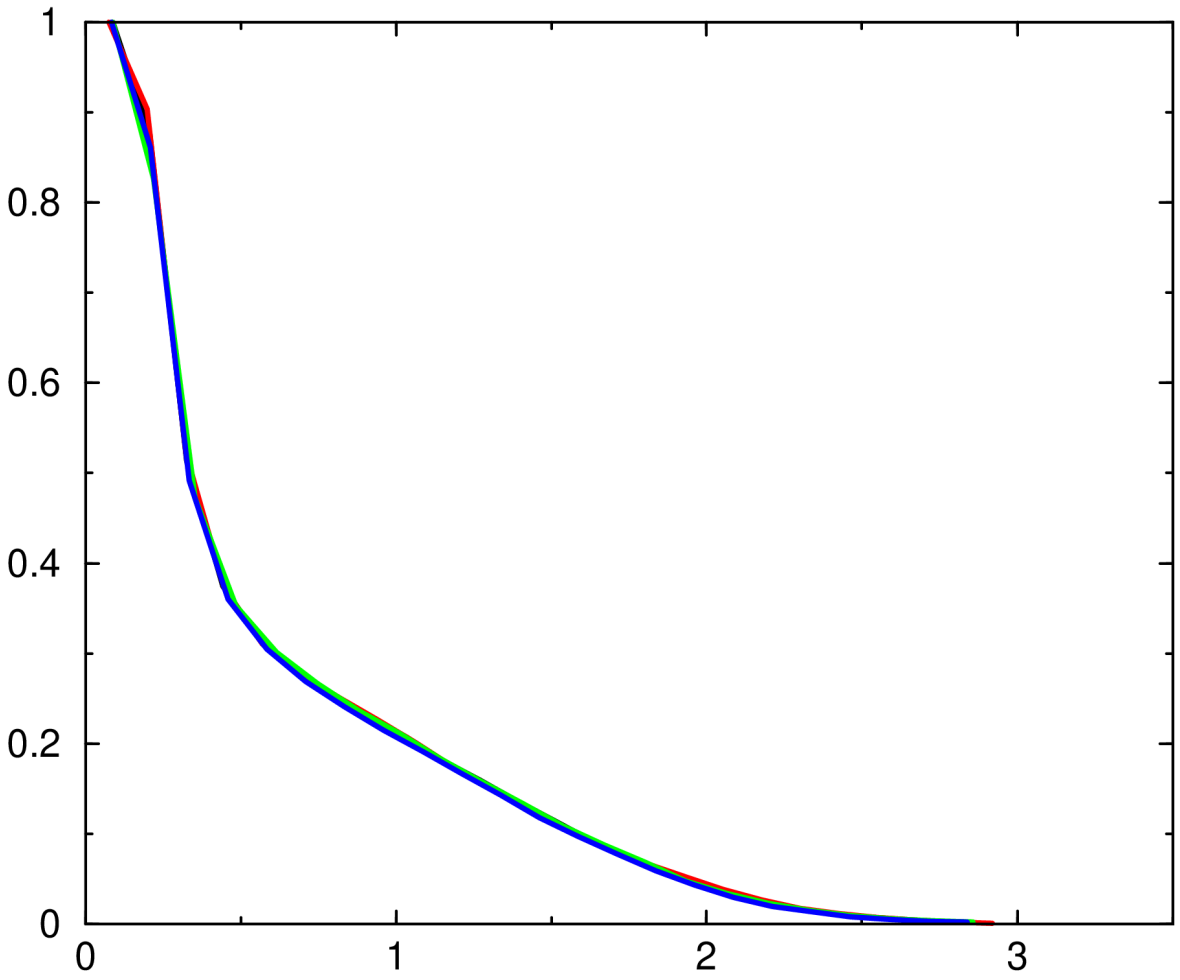

Fig.12 Comparison between the results obtained using a 30050x30050 matrix (black line), a 20356x20356 matrix (red line), a 15025x15025 matrix (green line) and a 25241x25241 matrix (blue line) for the cln025 peptide structure.

The blue line is slightly lower (has a set of slightly lower values) than the rest of the lines in a few areas, but generally the lines overlap in such a way that it is hard to distinguish them.

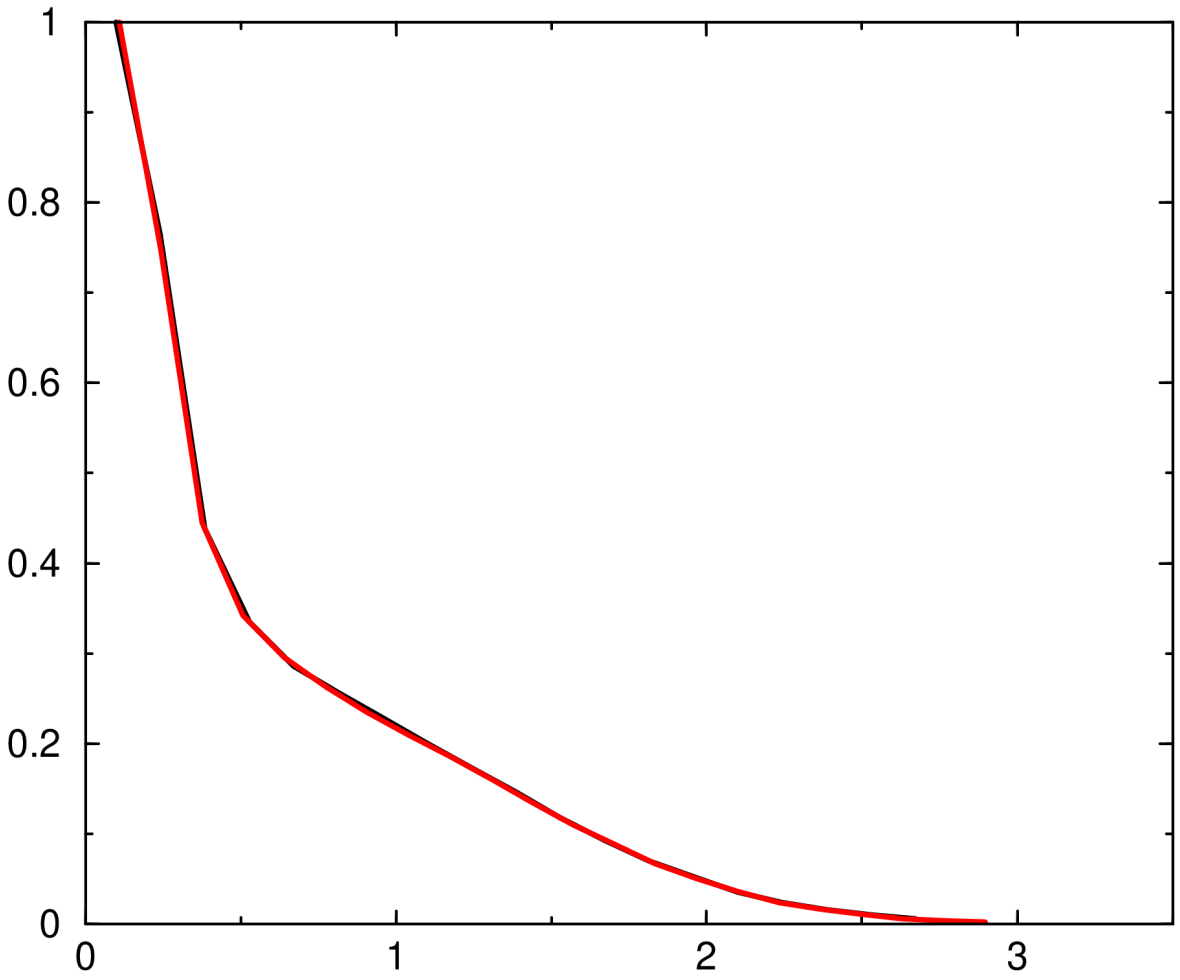

Fig.13 Comparison between the results obtained using a 14507x14507 matrix (black line) and a 10345x10345 matrix (red line) for the cln025 peptide structure.

The lines overlap almost perfectly.

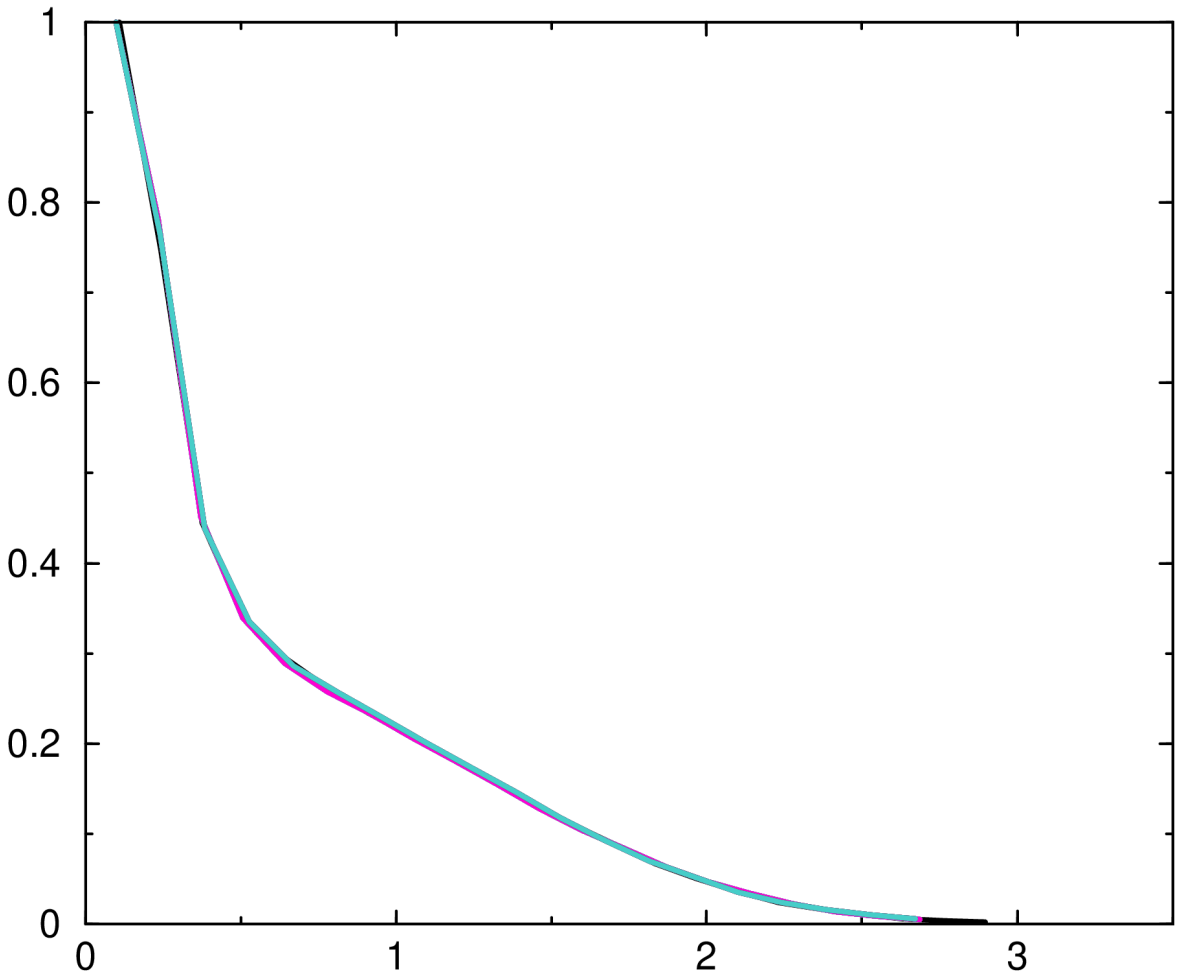

Fig.14 Comparison between the results obtained using a 10345x10345 matrix (black line), a 19702x19702 matrix (pink line) and a 14507x14507 matrix (light blue line) for the cln025 peptide structure.

The lines overlap almost perfectly. The light blue line overlaps a little more accurately to the black line in contrast to the pink line, which is a little lower (has a set of lower values) in a few areas of the graph.

The next graph shows the data that derive from both methods, for all the different matrices created for the evaluation of the classic “Good-Turing” and the “max of mins” method:

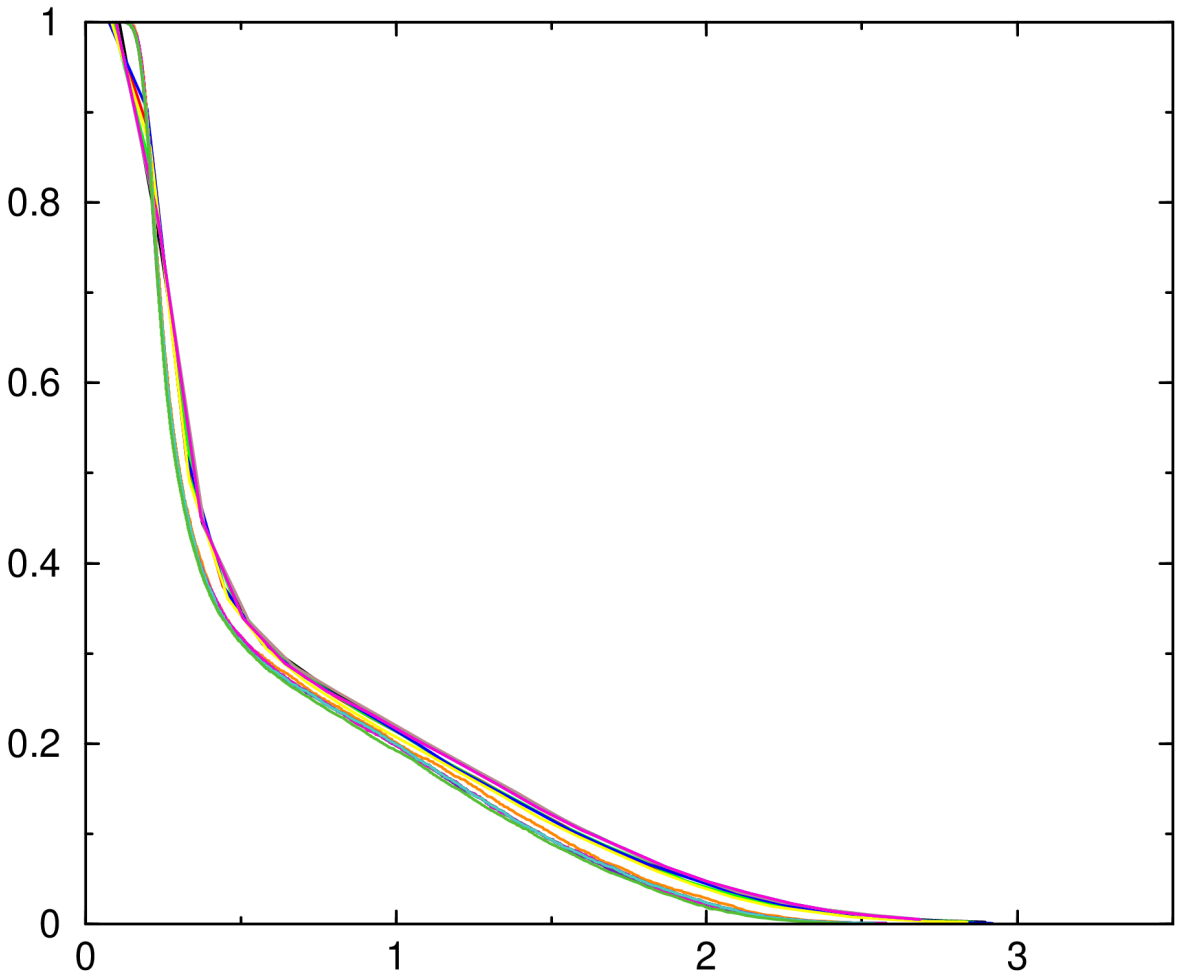

Fig.15 Comparison between the results obtained using the "max of mins" method for every different matrix created (lower set of lines) and the "Good-Turing" method for every different matrix created (upper set of lines), for the cln025 peptide structure.

The set of lines that are the result of the classic "Good-Turing" method are located more on the right in the graph compared to the set of lines deriving from the "max of mins" method. The set of lines that are the result of the classic "Good-Turing" method are very close to each other, as well as the set of lines from the "max of mins" method. The sets of lines for the two methods are close to each other but not so, that it is not distinct which line belongs to which set. The most ideal scenario would have these two sets overlapping. preventing clear distinction pertaining which line represents the results of each method. As it

appears, that only happens, for the lines of each set individually, meaning that the lines, which are the result of the classic “Good-Turing” method, overlap with each other in such way that it is not easy to tell each line from the other and in the results deriving from the “max of mins” method, its lines are overlapping with each other, again, in such a way that they cannot be efficiently analysed as separate lines in this graph.

2n0x:

The matrices created for the 2n0x peptide structure and whose data are being analysed here have been created by increasing the step (stride) between frames in order to obtain smaller sized matrices. The steps (strides) between frames that have been opted for each matrix for the purposes of this research are the following:

Stride (step) between frames	Matrix size
750	29834x29834 matrix
895	25000x25000 matrix
1100	20341x20341 matrix
1490	15017x15017 matrix
2180	10264x10264 matrix

Table 4. Step (stride) between frames & matrix size of the samples for the 2n0x peptide structure.

The optimal (sub-sampling) factors of the samples do not consistently decrease as the matrix size gets smaller as it is evident in Table 5.:

Optimal (sub-sampling) factor of:	Matrix size
8	10264x10264 matrix
9	15017x15017 matrix
24	20341x20341 matrix
37	25000x25000 matrix
32	29834x29834 matrix

Table 5. Optimal (sub-sampling) factor & matrix size of the samples for the 2n0x peptide structure.

In order to illustrate how robust the “max of mins” method is and how reliable smaller matrices are, graphs have been created out of the datasets obtained from the different runs.

When it comes to the data deriving from the classic "Good-Turing" method, for the 2n0x peptide structure, the graphs are the following:

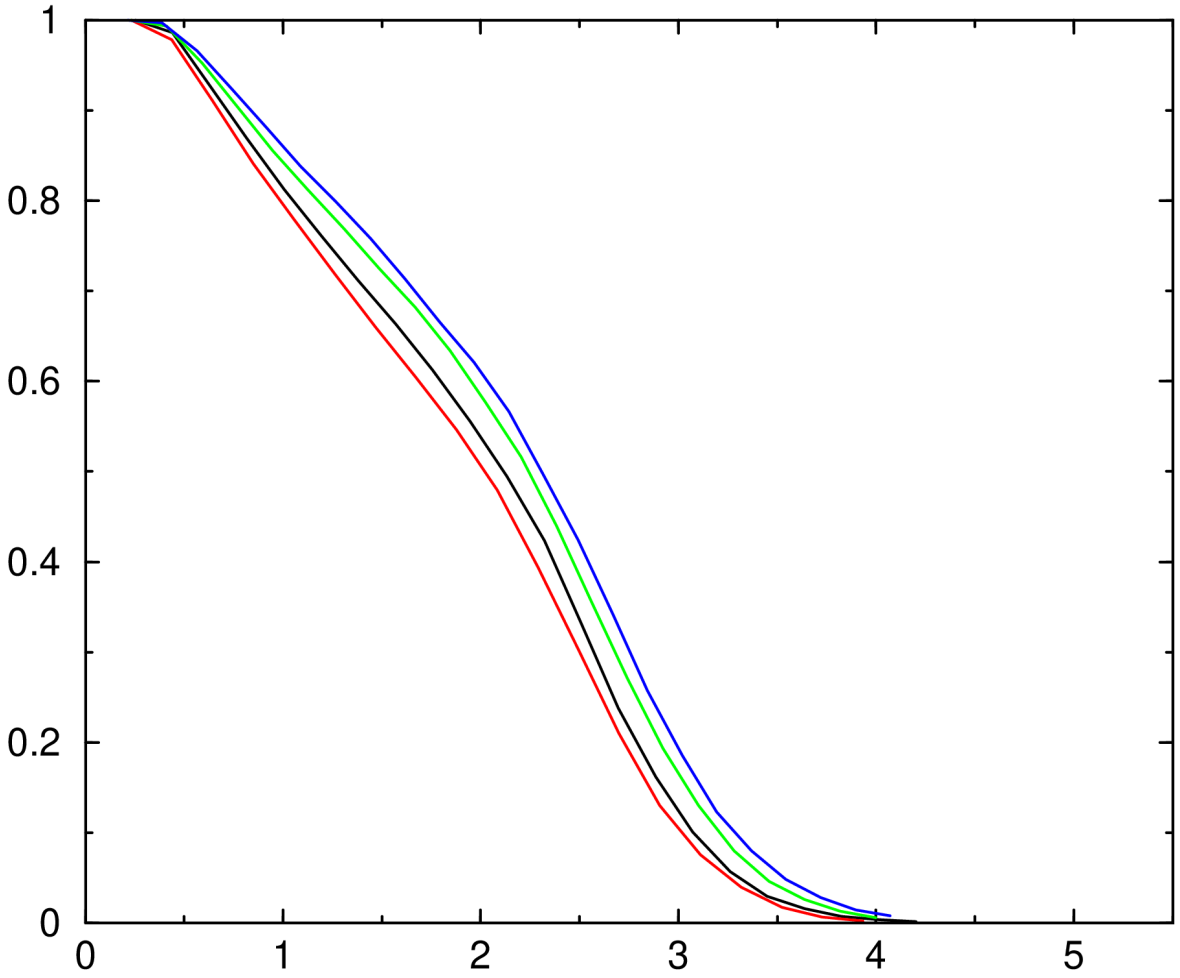

Fig.16 Comparison between the results obtained using a 10264x10264 matrix (black line), a 15017x15017 matrix (red line), a 20341x20341 matrix (green line) and a 25000x25000 matrix (blue line) for the 2n0x peptide structure.

The lines do not overlap and are relatively close. More specifically, the black and red lines are closer with each other, than with the other lines of the graph and in a similar fashion, the green and blue lines are closer with each other, than with the other lines of the graph. The red line, which derives from a larger matrix (15017x15017 matrix), than the black line (10264x10264 matrix) appears to be more on the "left" side of the graph (meaning that it appears to have a lower set of values than the black line).

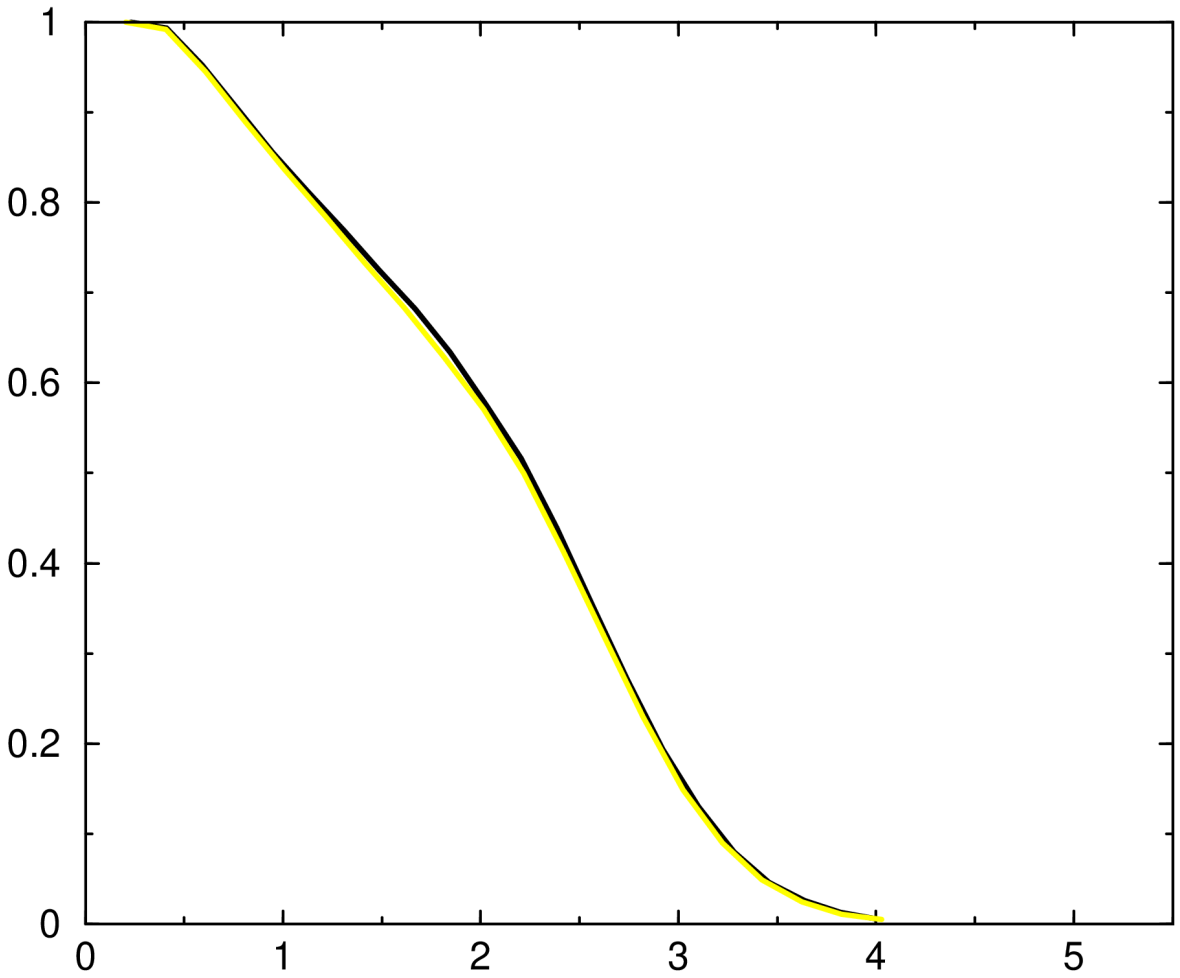

Fig.17 Comparison between the results obtained using a 20341x20341 matrix (black line) and a 29834x29834 matrix (yellow line) for the 2n0x peptide structure.

The lines overlap almost perfectly. The biggest non overlapping areas between these two lines are observed in the lower RMSD values where the probability values are always higher.

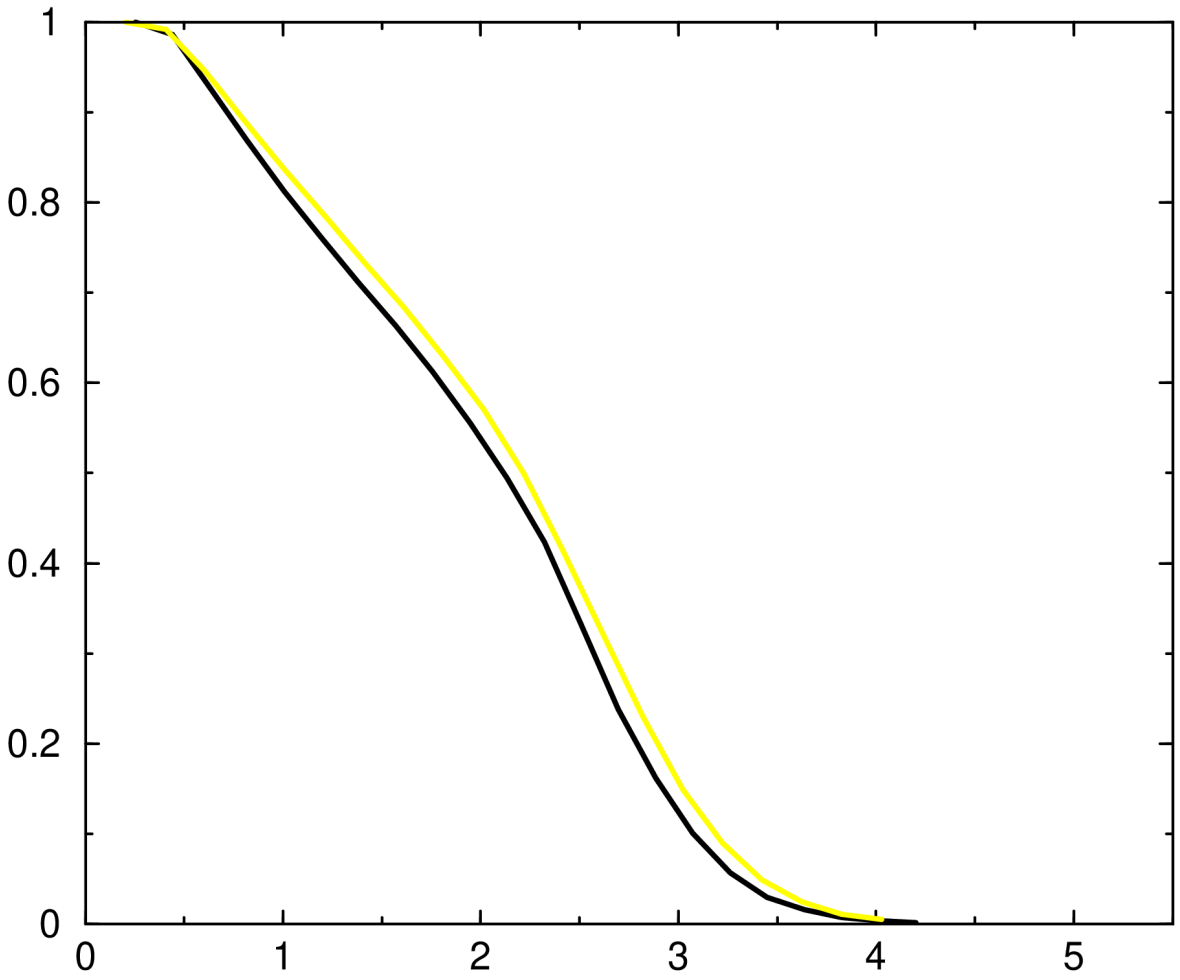

Fig.18 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 29834x29834 matrix (yellow line) for the 2n0x peptide structure.

The lines are very close, but they do not overlap in almost any area of the graph.

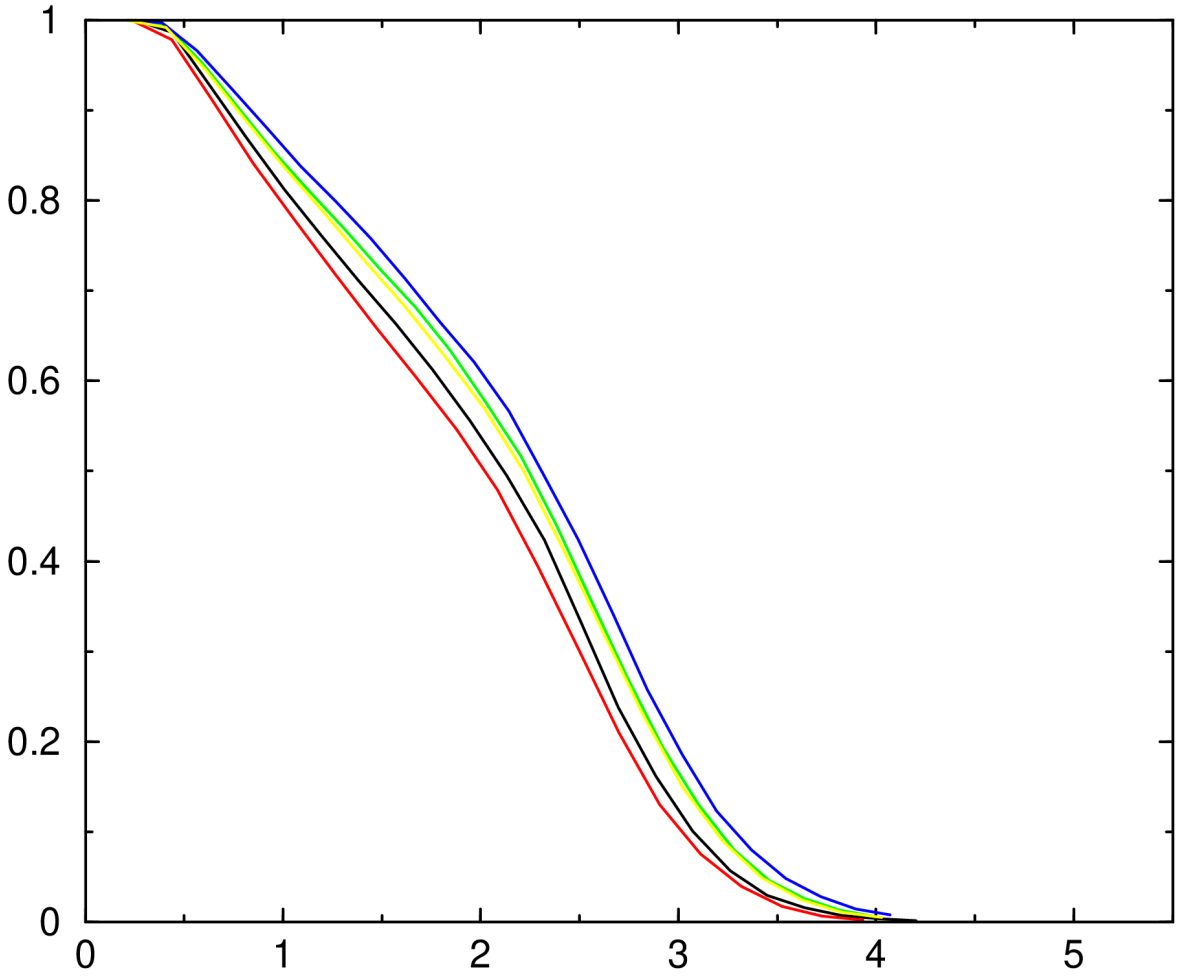

Fig.19 Comparison between the results obtained using a 10264x10264 matrix (black line), a 15017x15017 matrix (red line), a 20341x20341 matrix (green line), a 25000x25000 matrix (blue line) and a 29834x29834 matrix (yellow line) for the 2n0x peptide structure.

As it has already been noted in Fig.17, the green and the yellow lines overlap almost perfectly. The blue line does not represent the bigger matrix, but it is the one located more on the “right” of the graph (meaning that it has a higher set of values).

The graphs deriving from the “max of mins” method for the 2n0x peptide structure are the following:

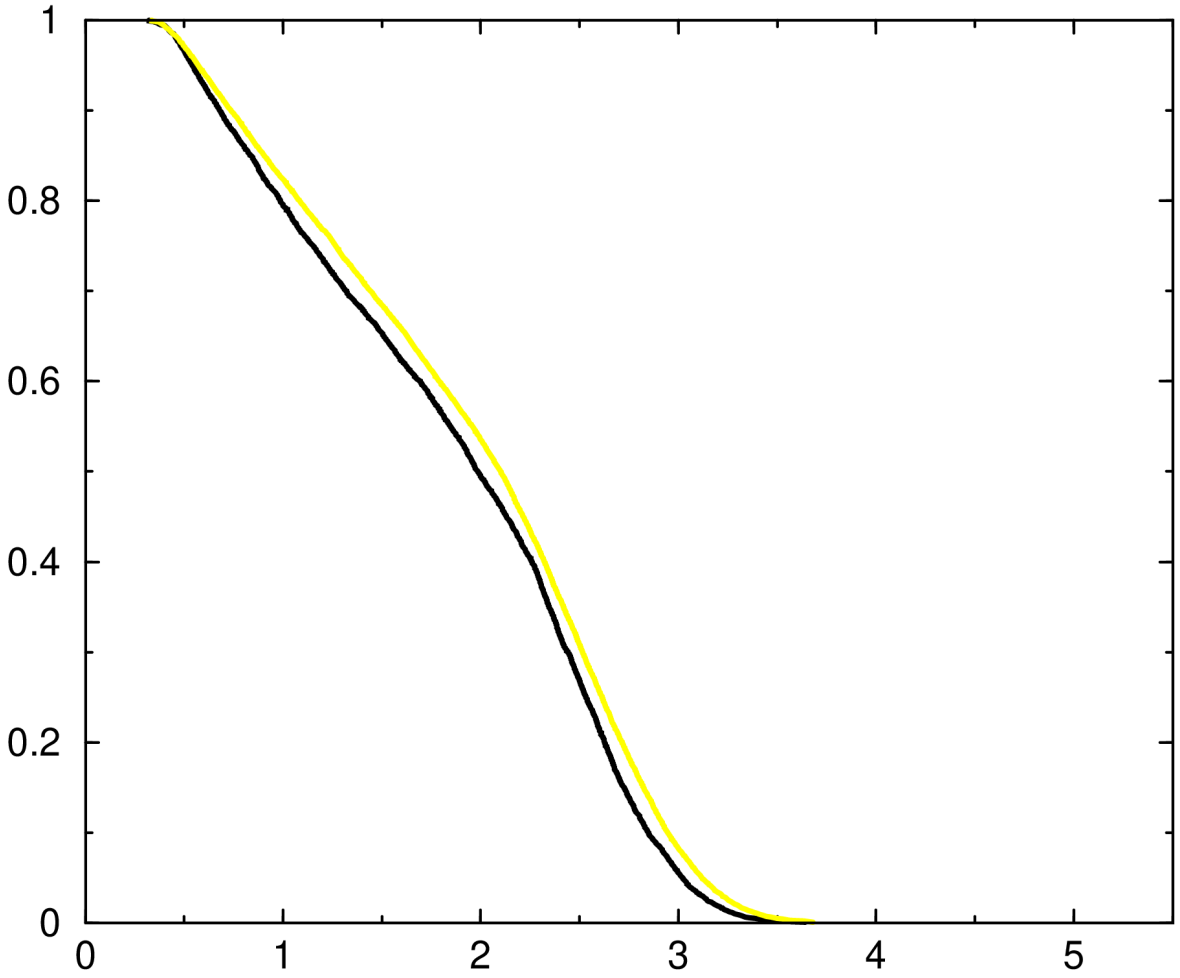

Fig.20 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 29834x29834 matrix (yellow line) for the 2n0x peptide structure.

The lines are very close to each other, but they do not overlap almost anywhere. The lines presented in Fig.18 represent the same matrices, but the lines in that graph (Fig.18) derive from the classic “Good-Turing” method and they are not as close with each as the ones in this graph (Fig.20).

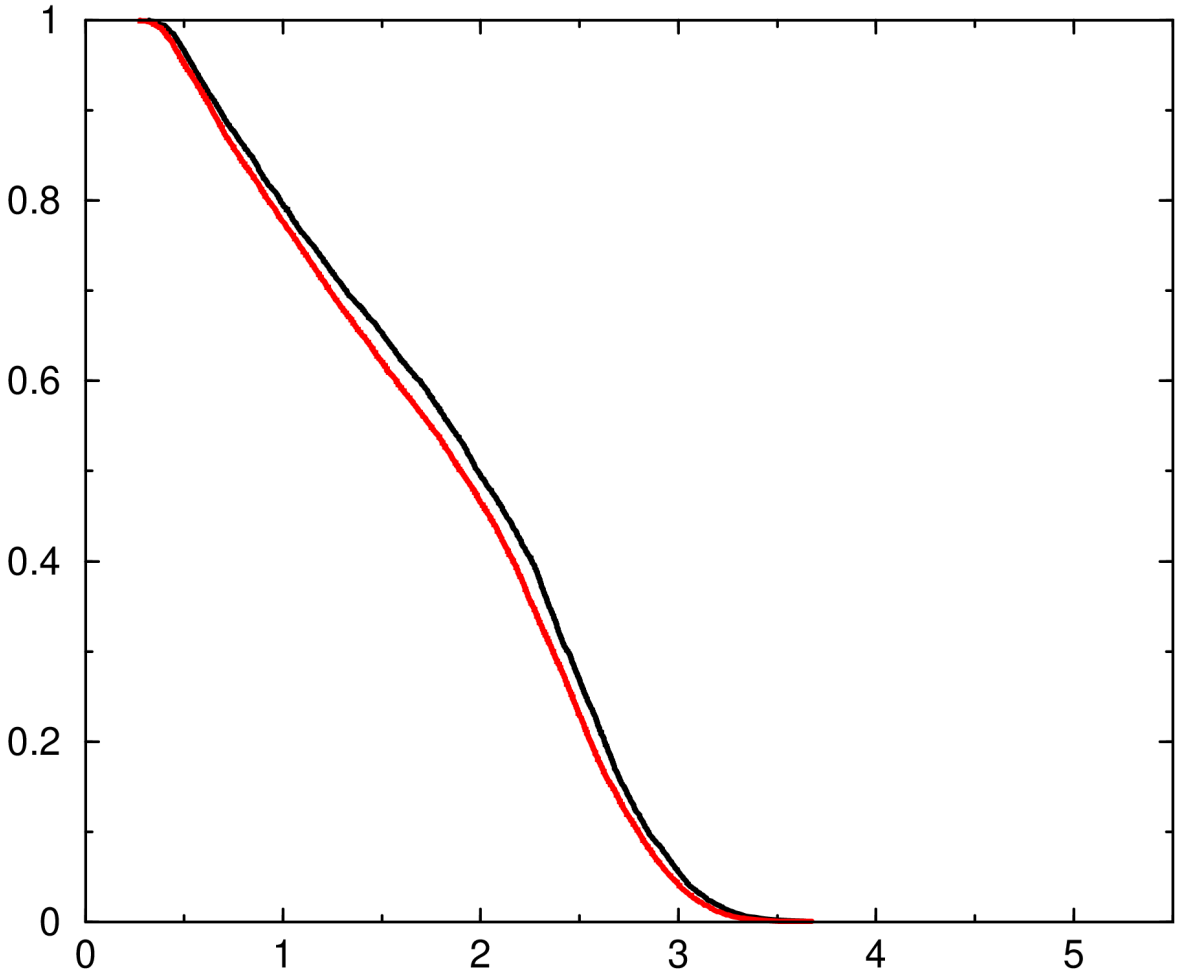

Fig.21 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 15017x15017 matrix (red line) for the 2n0x peptide structure.

The lines are very close, but they do not overlap almost anywhere. The red line, which is the resulting line for the 15017x15017 matrix, is located more on the "left" than the black line which derives from the 10264x10264 matrix. The same exact observation regarding the way the lines are placed in the graph applies for Fig.16 and Fig.19, where the lines in these graphs derive from the classic "Good-Turing" method, but for the same sized matrices (10264x10264 matrix and 15017x15017 matrix). The green line looks almost between the red and the black lines. More specifically, the green line appears to be a little closer to the black line at lower RMSD values and a little closer to the red line at higher RMSD values.

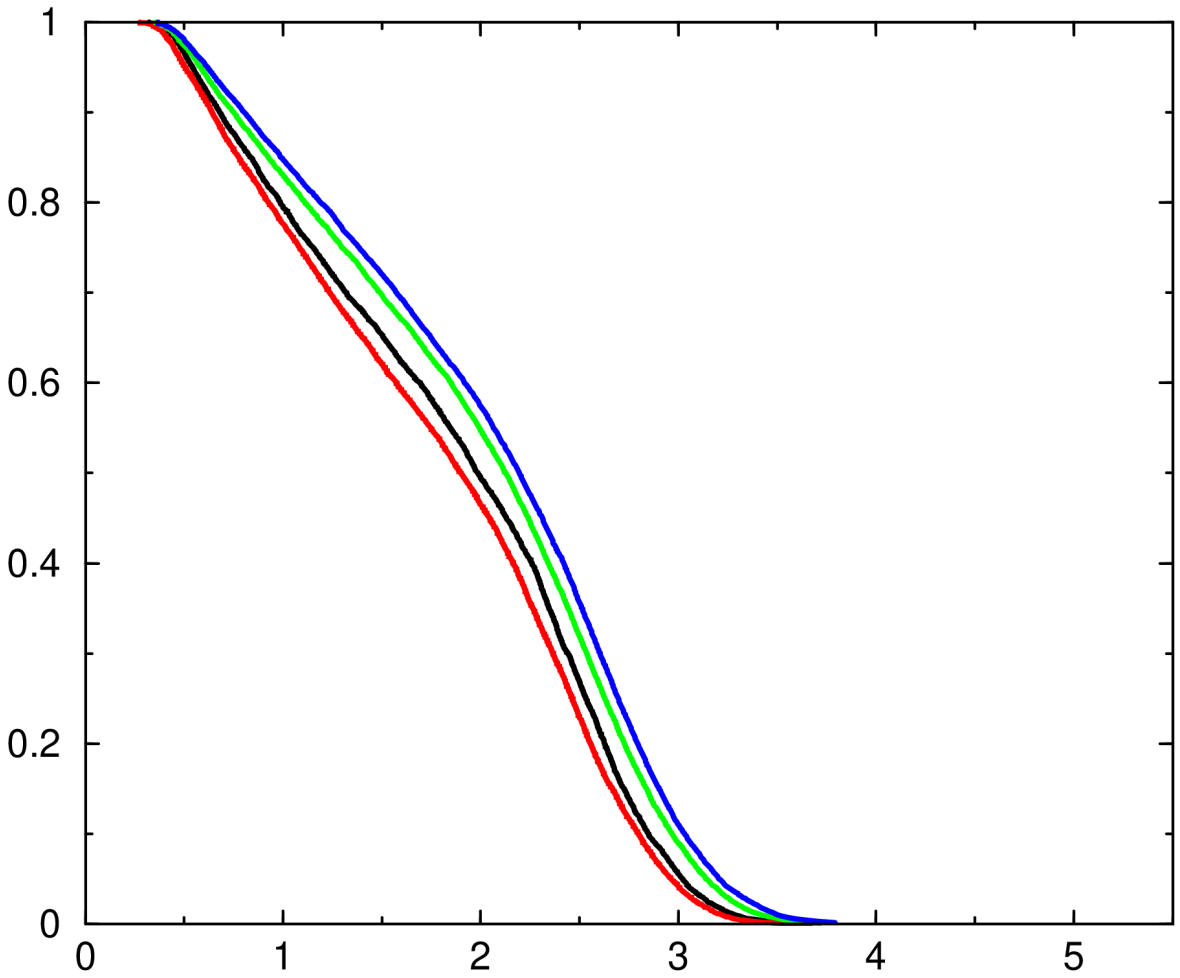

Fig.22 Comparison between the results obtained using a 10264x10264 matrix (black line), a 15017x15017 matrix (red line), a 20341x20341 matrix (green line) and a 25000x25000 matrix (blue line) for the 2n0x peptide structure.

The lines are close to each other. The “pairs” of the red and black line and the green and blue line are very close to each other. The graph that represents the lines for the same matrices, but is the result of the classic “Good-Turing” method is Fig.16 and although the lines are not located where the lines that have occurred from the “max of mins” method (Fig.22) are, the left to right “row” of the lines is the same.

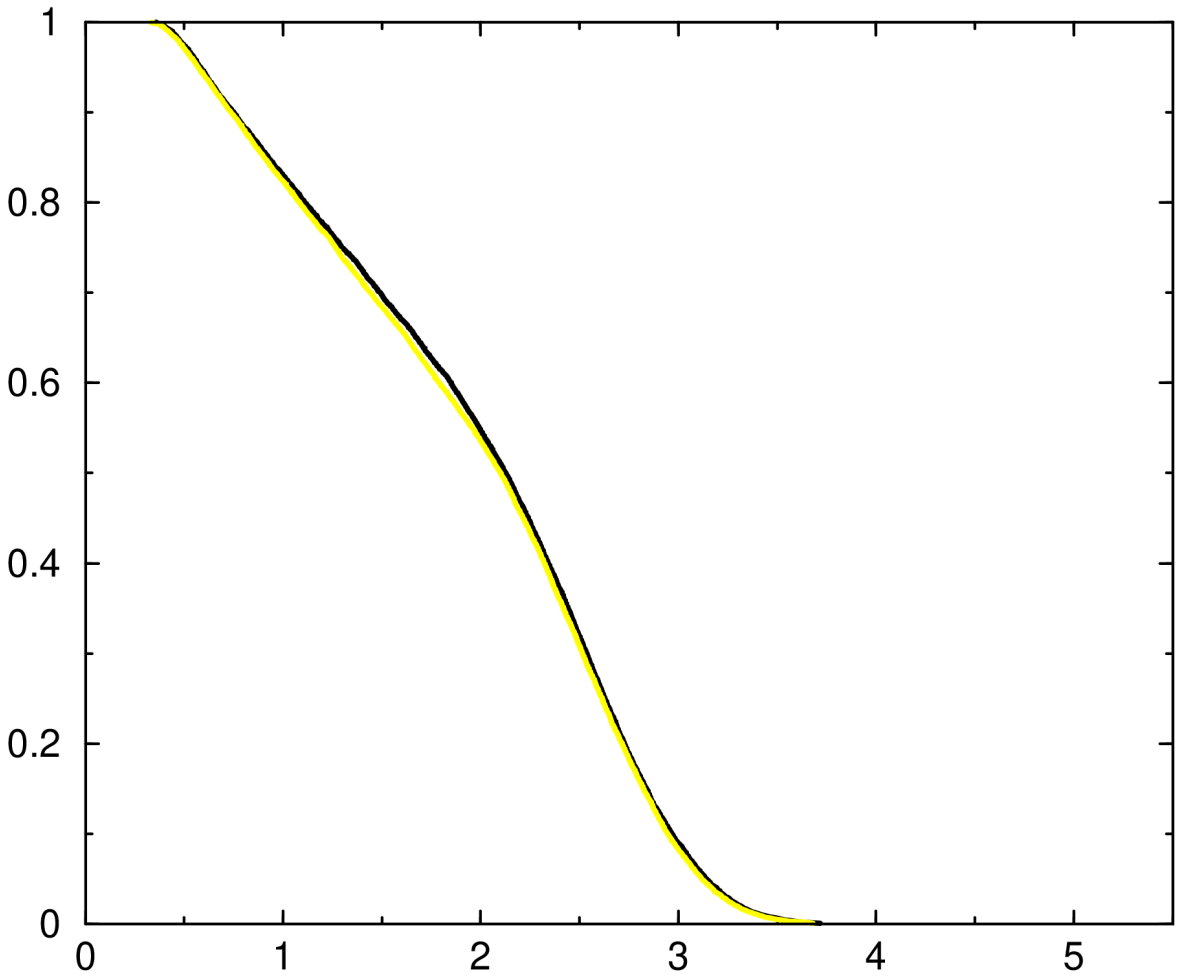

Fig.23 Comparison between the results obtained using a 20341x20341 matrix (black line) and a 29834x29834 matrix (yellow line) for the 2n0x peptide structure.

The lines overlap almost perfectly, almost everywhere. The area where the lines do not overlap is mainly between the RMSD values of 1-2 Å.

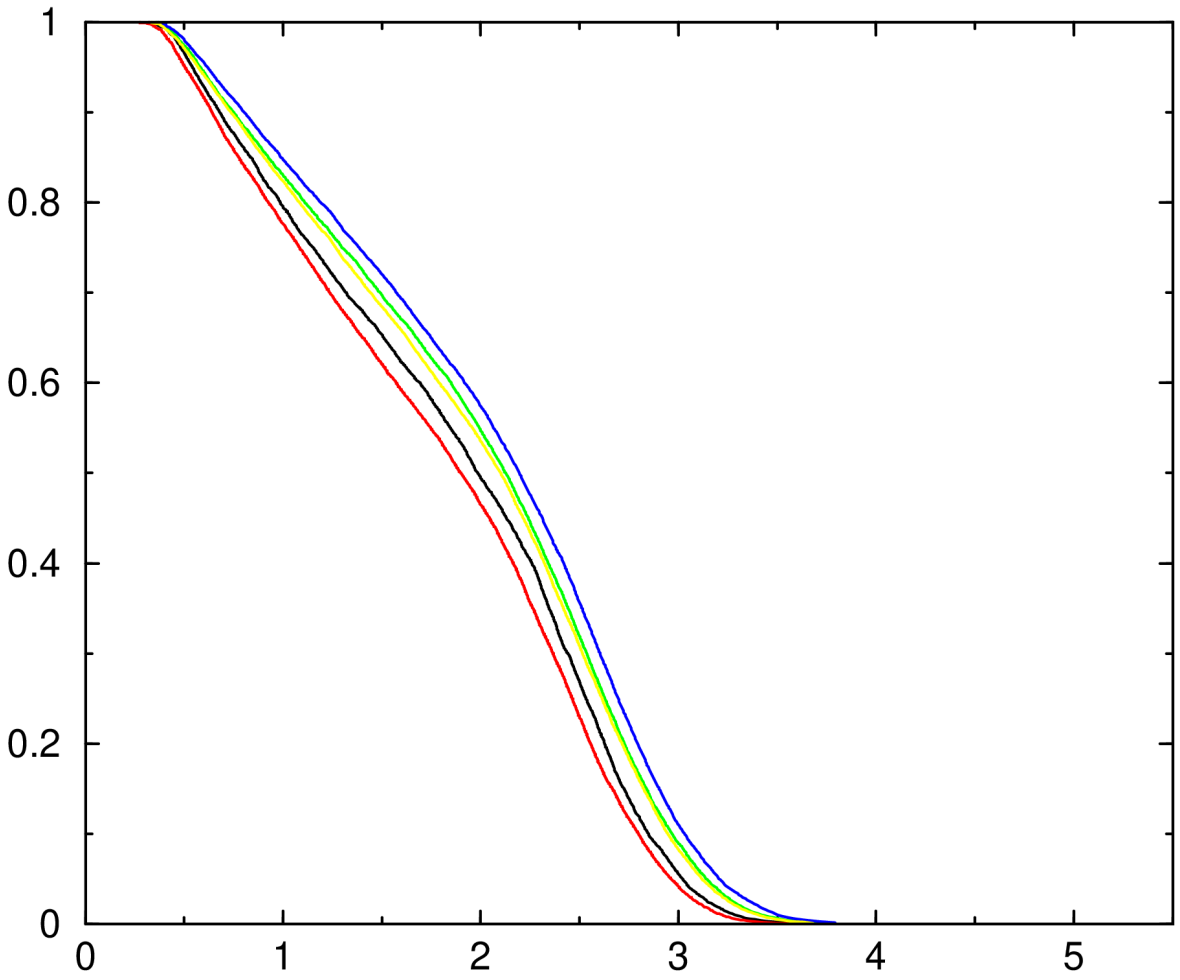

Fig.24 Comparison between the results obtained using a 10264x10264 matrix (black line), a 15017x15017 matrix (red line), a 20341x20341 matrix (green line), a 25000x25000 matrix (blue line) and a 29834x29834 matrix (yellow line) for the 2n0x peptide structure.

As it has already been noted in Fig.23, the green and the yellow lines overlap almost perfectly. In this figure (Fig.24), the blue line does not represent the bigger matrix, but it is the one located more on the “right” of the graph (meaning it has a higher set of values).

The next graph shows the data that derive from both methods, for all the different matrices created for the evaluation of the classic “Good-Turing” and the ‘max of mins” method (for the 2n0x peptide structure):

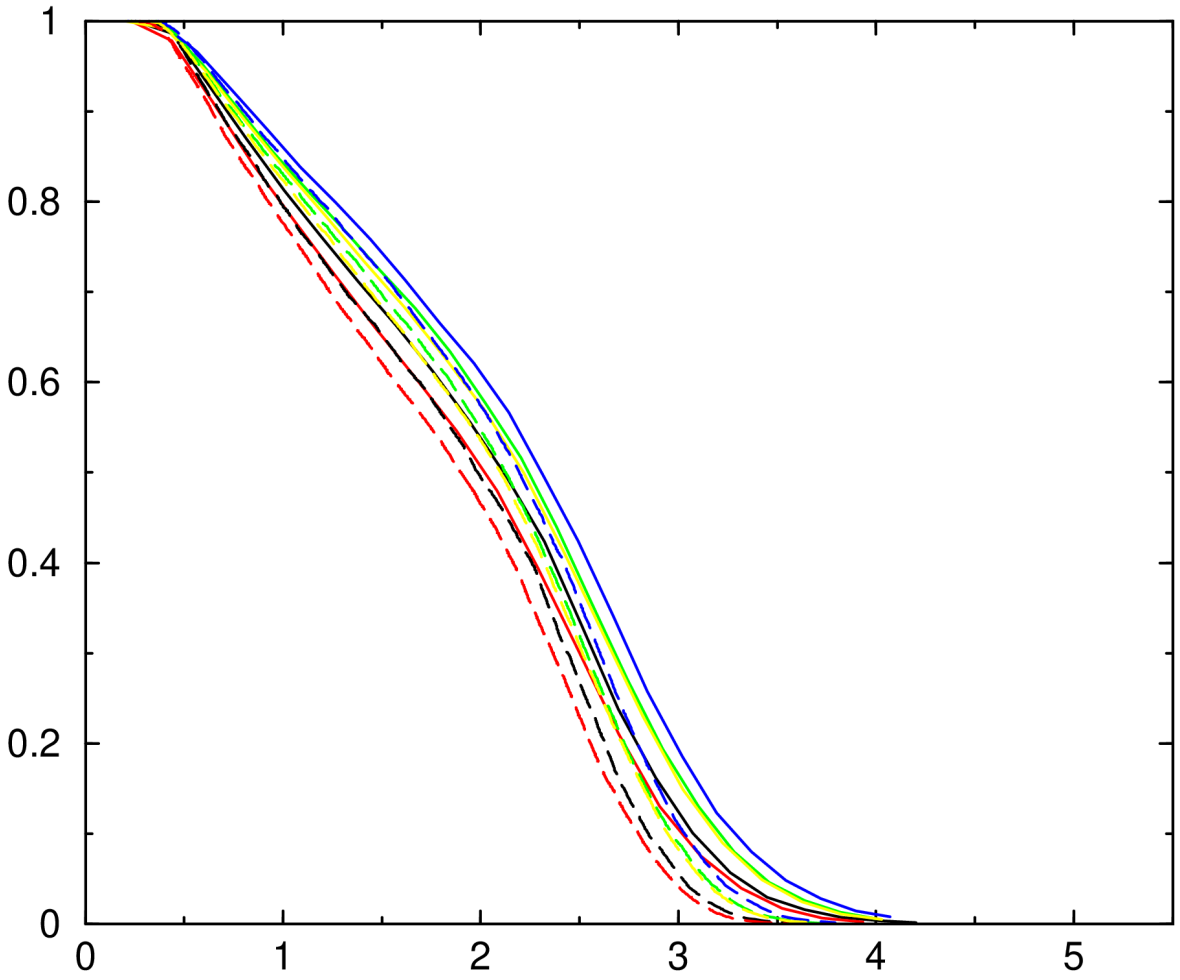

Fig.25 Comparison between the results obtained using the “max of mins” method for every matrix (dashed set of lines) and the “Good-Turing” method for every matrix (solid set of lines) for the 2n0x peptide structure.

The lines from each of the two methods are not very close to each other. Regarding the lines deriving from the “max of mins” method, the shift towards the “left” on the graph is obvious.

3.2 The “independent” method

Having noted above in this section, the reason why the trajectories of the peptide structures of cln025 and 2n0x were used as samples for the evaluation of the “independent” method’s efficacy, here follow the results that occurred from the runs of the “independent” method for various steps (strides) between the frames.

cln025:

In the following graphs are presented the results that derive from the classic "Good-Turing" method (black lines) and the "independent" method (scatter plots) for the cln025 peptide structure.

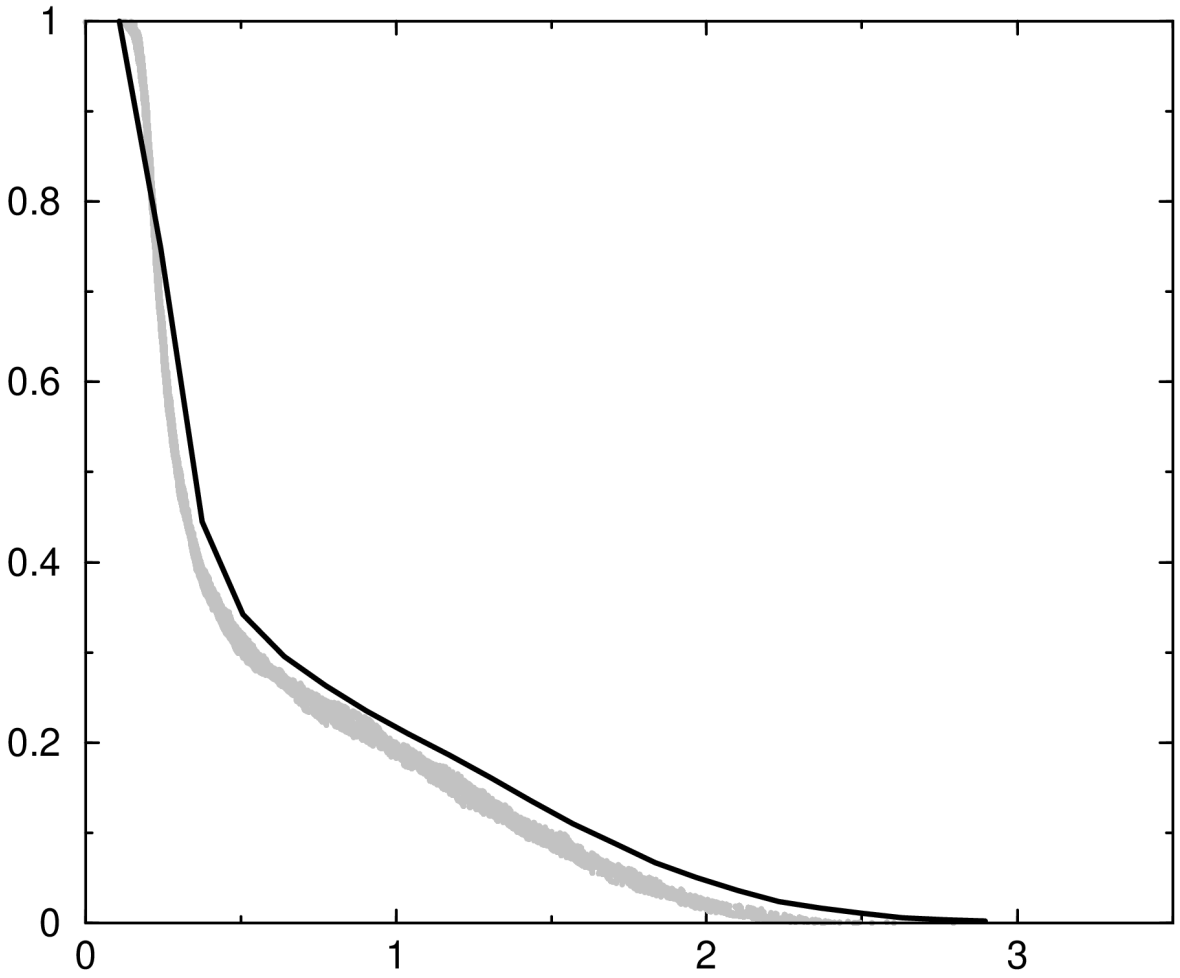

Fig.26 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 5K scatter plot (gray scatter plot) for the cln025 peptide structure.

The scatter plot-which is the result produced by the "independent" method-is close to the black line, but they do not overlap in almost any area of the graph. The scatter plot is more on the "left" of the graph (has a lower set of values), with the exception being the 0-0.25Å area.

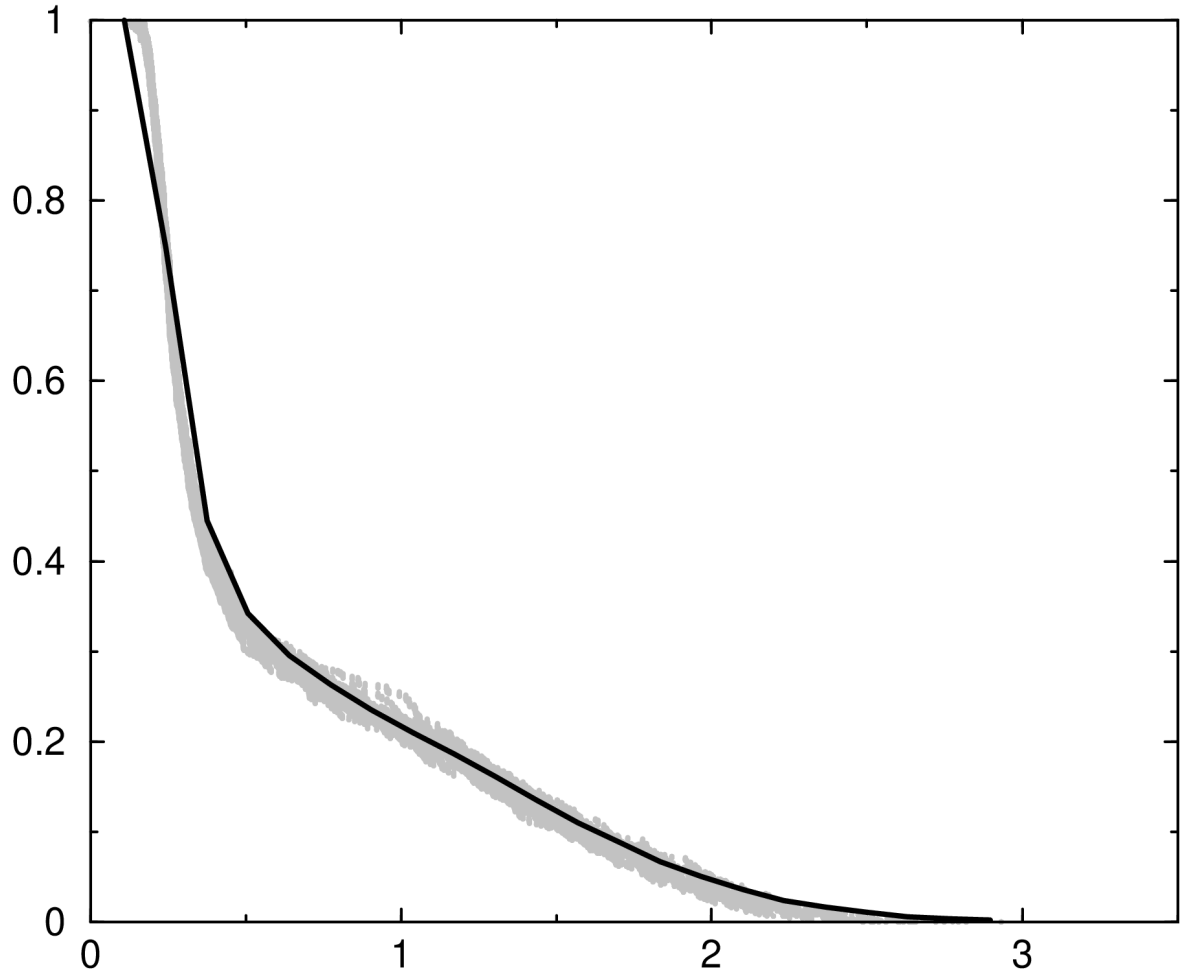

Fig.27 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 10K scatter plot (gray scatter plot) for the cln025 peptide structure.

It is obvious that the 5K increase in the step has resulted in a better scatter plot than the one in Fig.26. The line and the scatter plot overlap in many areas of the graph, but the scatter plot is placed slightly lower (has a set of slightly lower values).

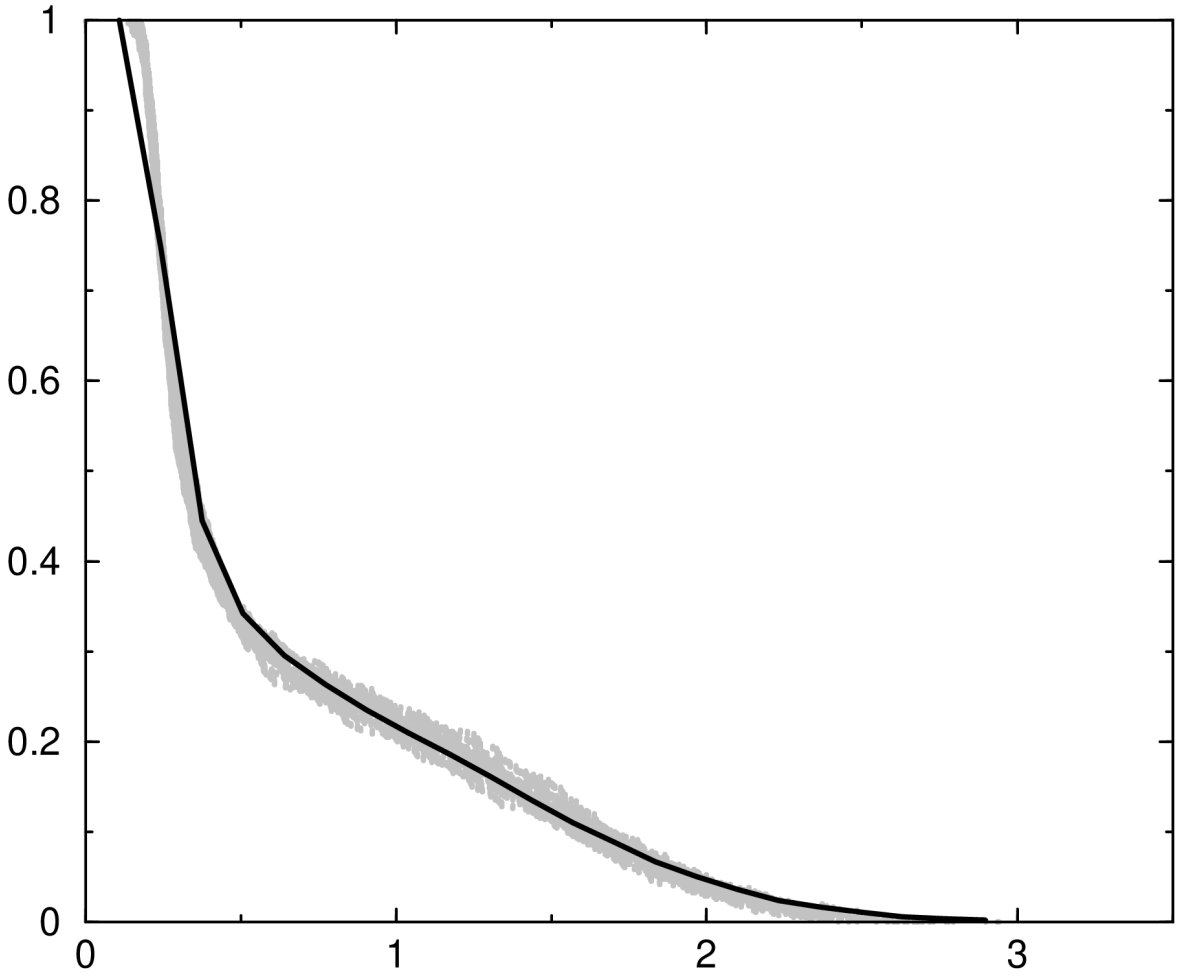

Fig.28 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 11.5K scatter plot (gray scatter plot) for the cln025 peptide structure.

The line and the scatter plot overlap in many areas of the graph and in many of these areas, the line appears to be “in the middle” of the scatter plot, which is the desired outcome.

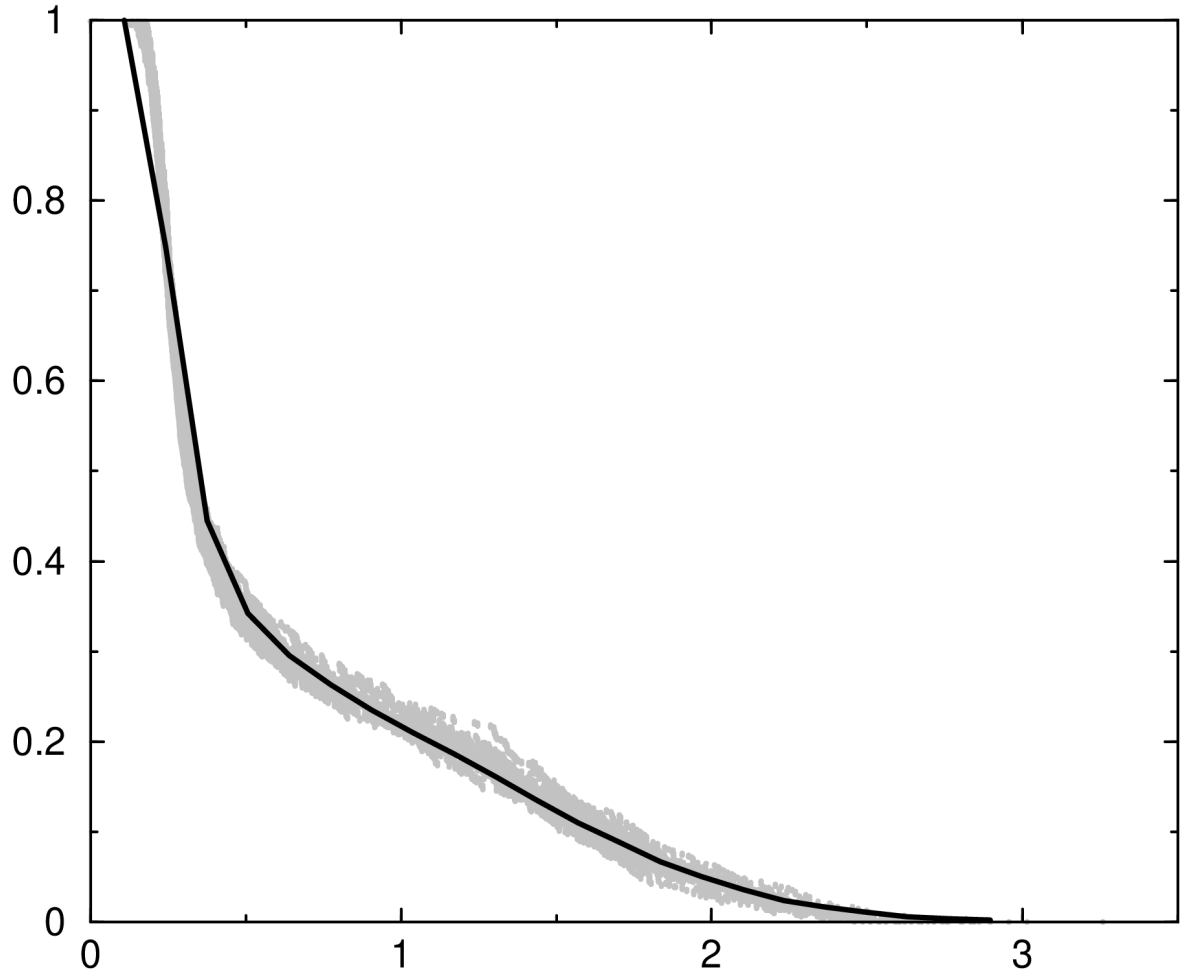

Fig.29 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 12K scatter plot (gray scatter plot) for the cln025 peptide structure.

Just like Fig.28, the line and the scatter plot overlap in many areas of the graph, but the scatter plot has a slightly higher set of values than the line. When it comes to the values for the 1-2Å area, the line and the scatter plot overlap better in Fig.28, whereas for the 2-3Å area, the overlap of the line and the scatter plot is better than the one in Fig.28.

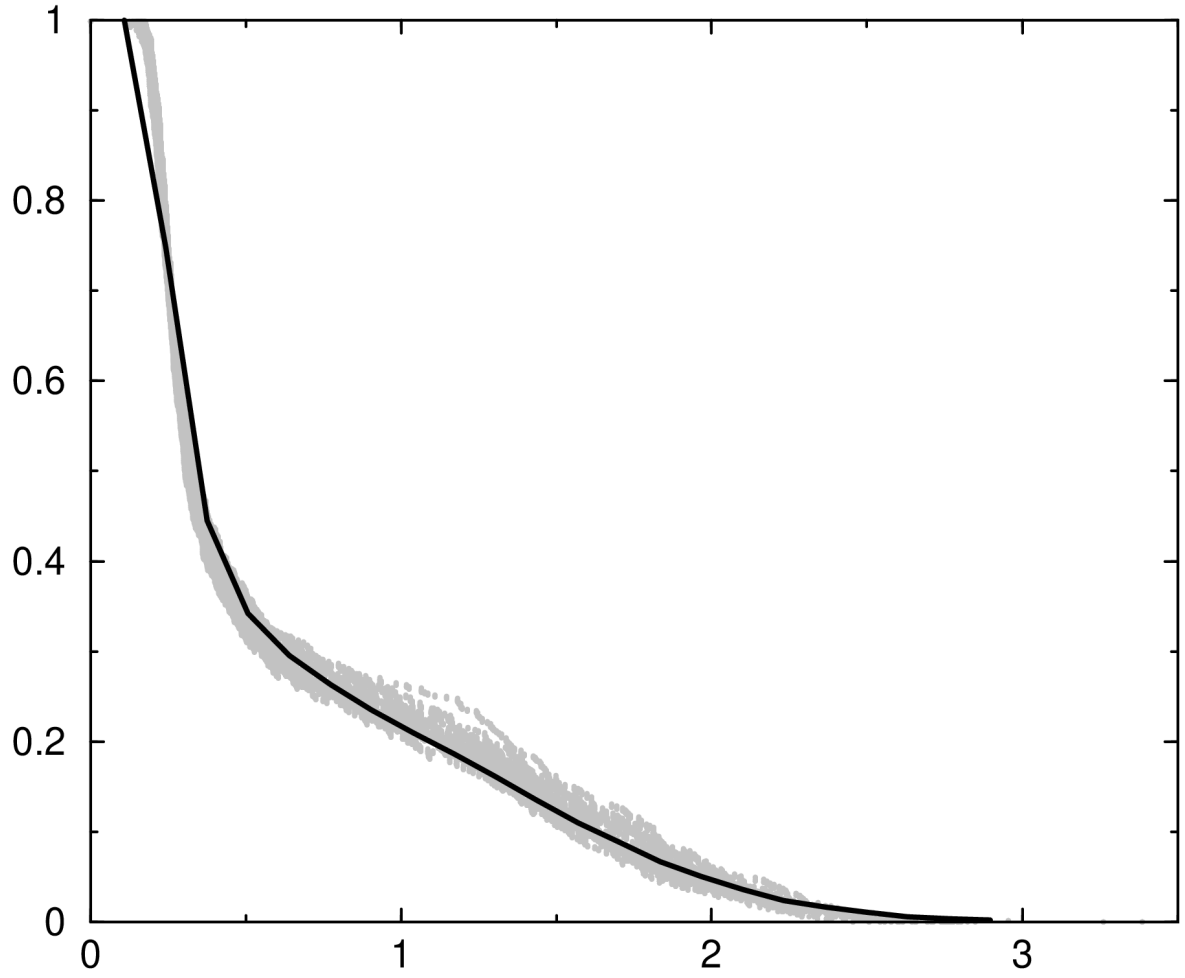

Fig.30 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 12.5K scatter plot (gray scatter plot) for the cln025 peptide structure.

As the step increases, the line and the scatter plot overlap, but the line is no longer “in the middle” of the scatter plot. The scatter plot has a higher set of values than the line in many areas that are very important for the evaluation of the result’s quality.

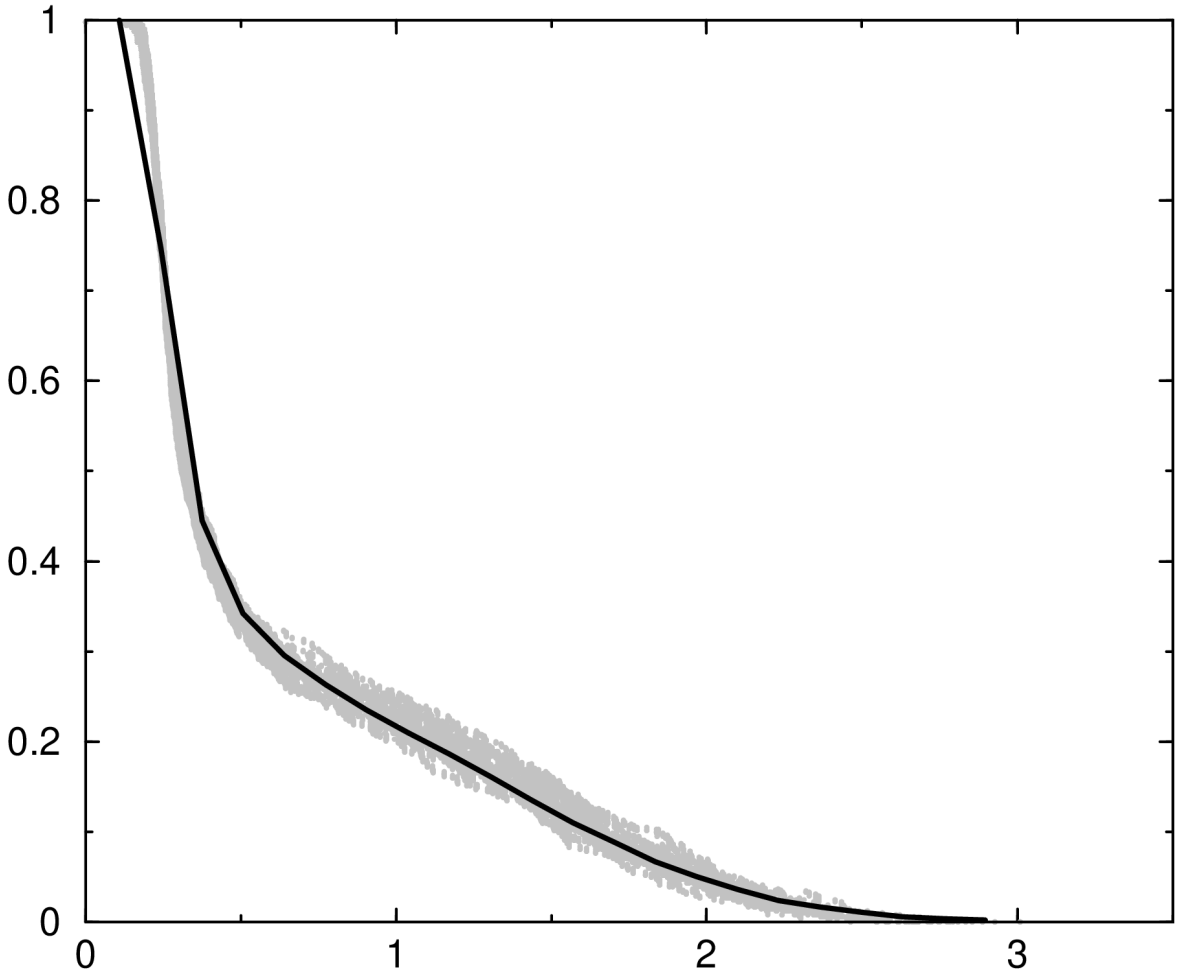

Fig.31 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 13K scatter plot (gray scatter plot) for the cln025 peptide structure.

The lines still overlap in many areas of the graph but the scatter plot has a higher set of values than the line for the most important RMSD values.

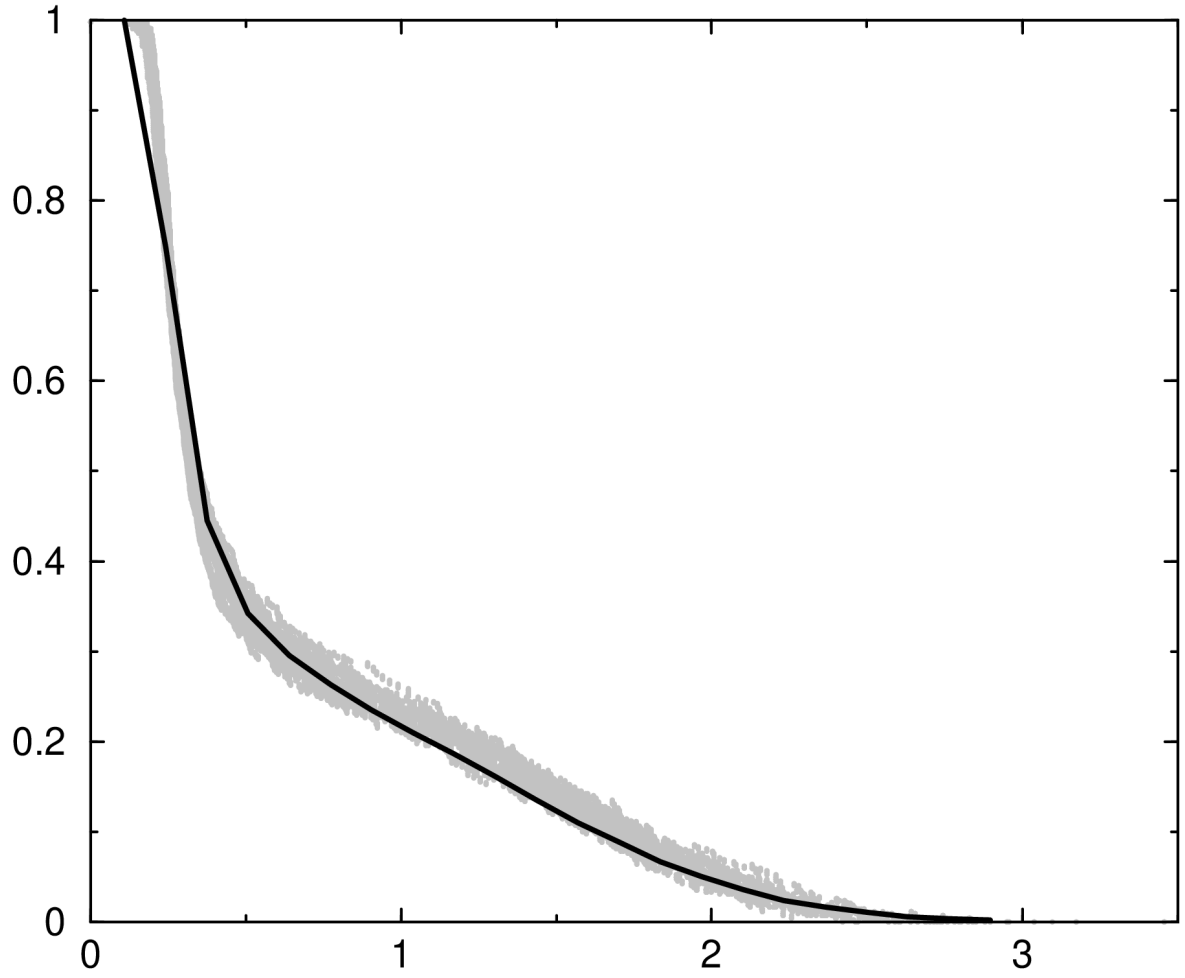

Fig.32 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 15K scatter plot (gray scatter plot) for the cln025 peptide structure.

The line overlaps with the scatter plot in almost every area of the graph, which is the case for Fig.28-31 as well, but the scatter plot is located higher (has a higher set of values) for the most important RMSD values.

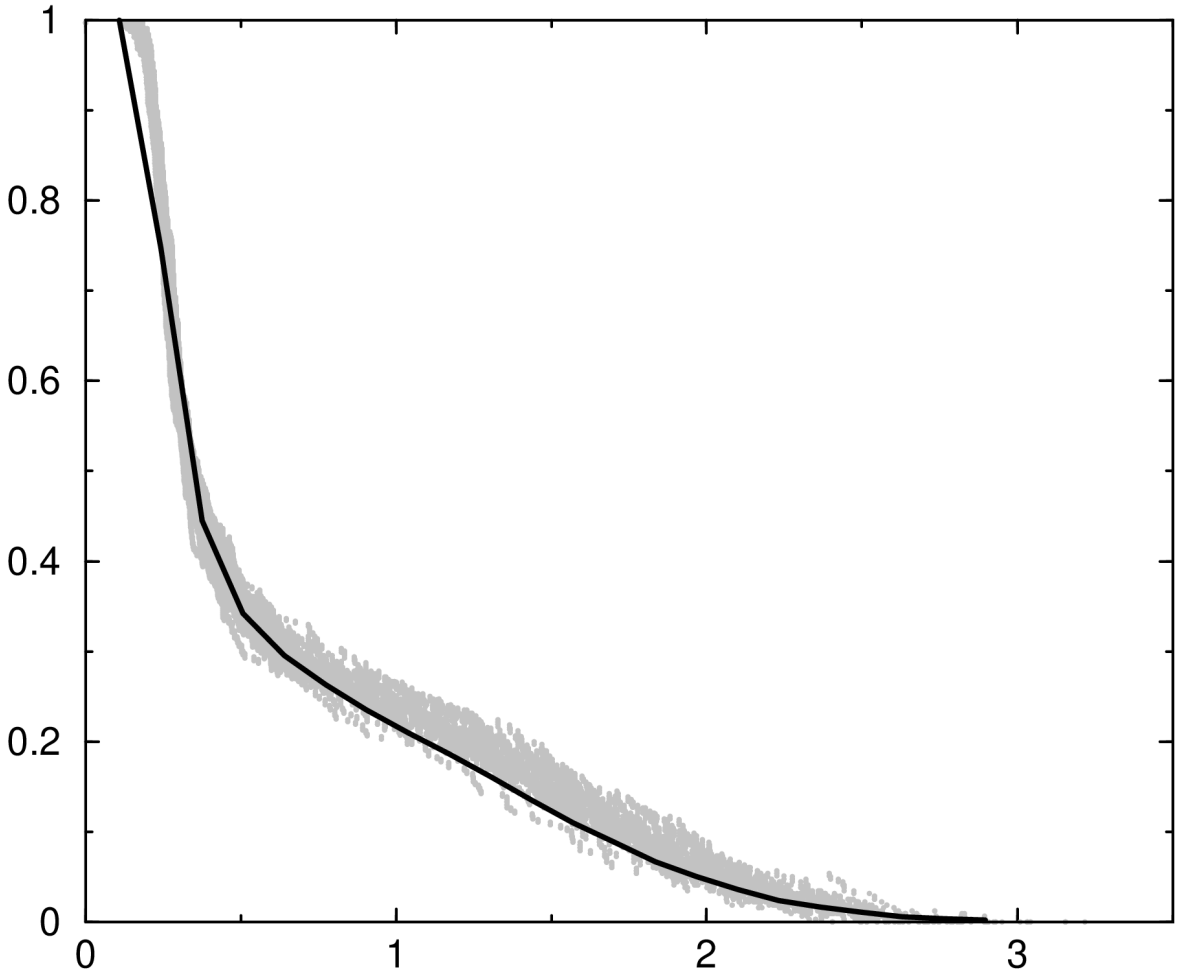

Fig.33 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 20K scatter plot (gray scatter plot) for the cln025 peptide structure.

As it has already been observed in Fig.32, the line and the scatter plot overlap in almost every area of the graph, but the scatter plot is higher (has a higher set of values) for the most important RMSD values.

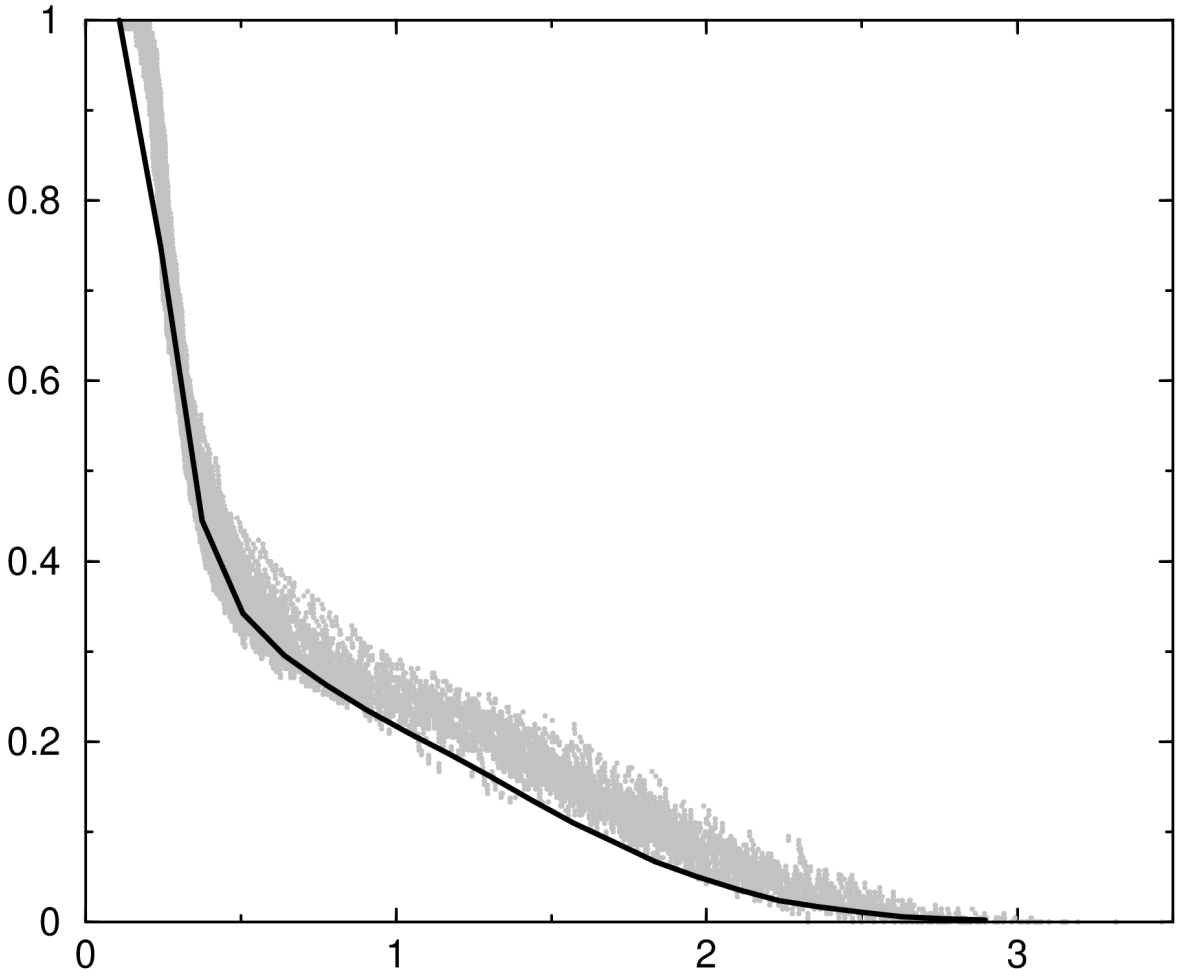

Fig.34 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 30K scatter plot (gray scatter plot) for the cln025 peptide structure.

The line and the scatter plot do not overlap in as many areas as they do in Fig.29-33, since the scatter plot is located even higher (has an even higher set of values).

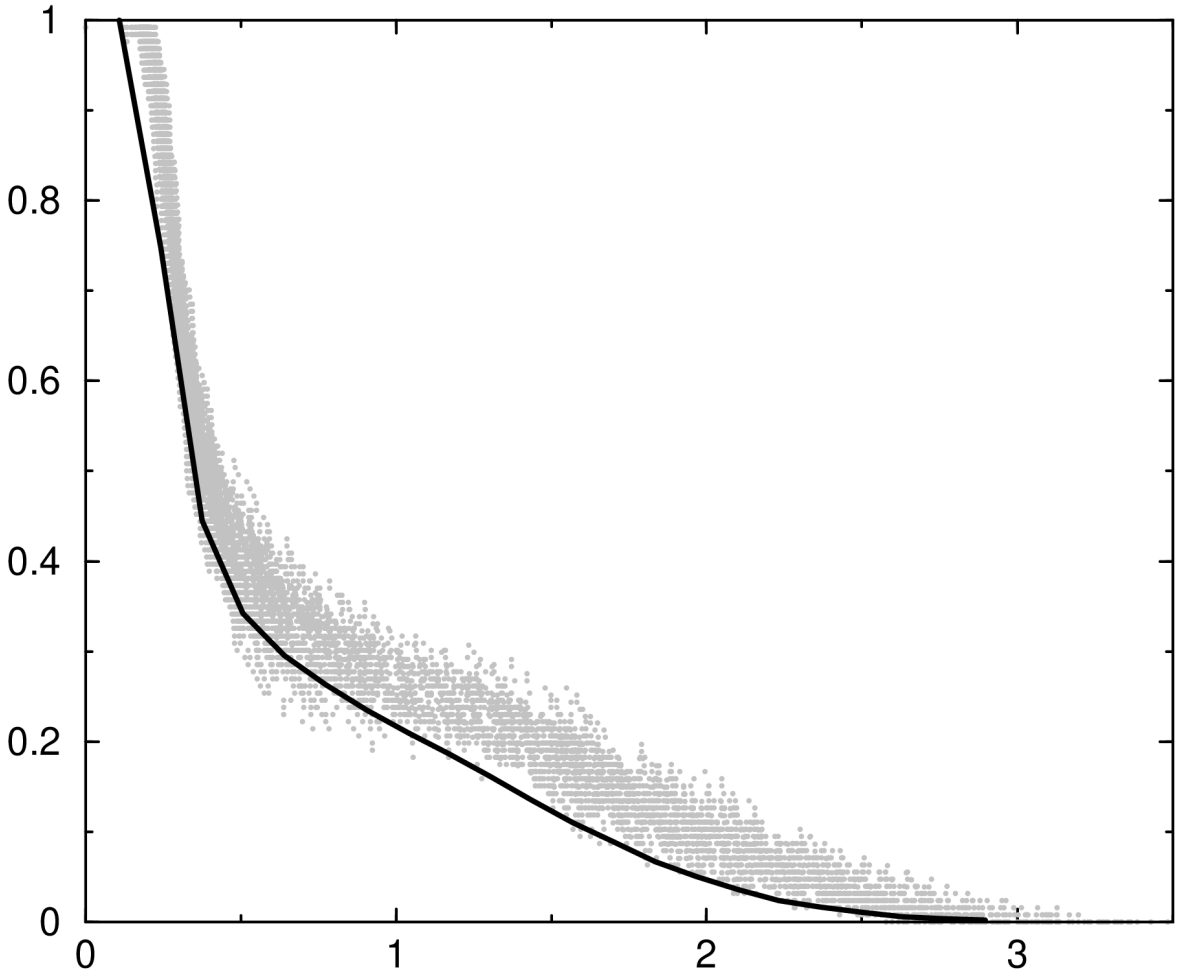

Fig.35 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 50K scatter plot (gray scatter plot) for the cln025 peptide structure.

The line and the scatter plot overlap in few areas of the graph. The scatter plot is higher (has a higher set of values) than the line.

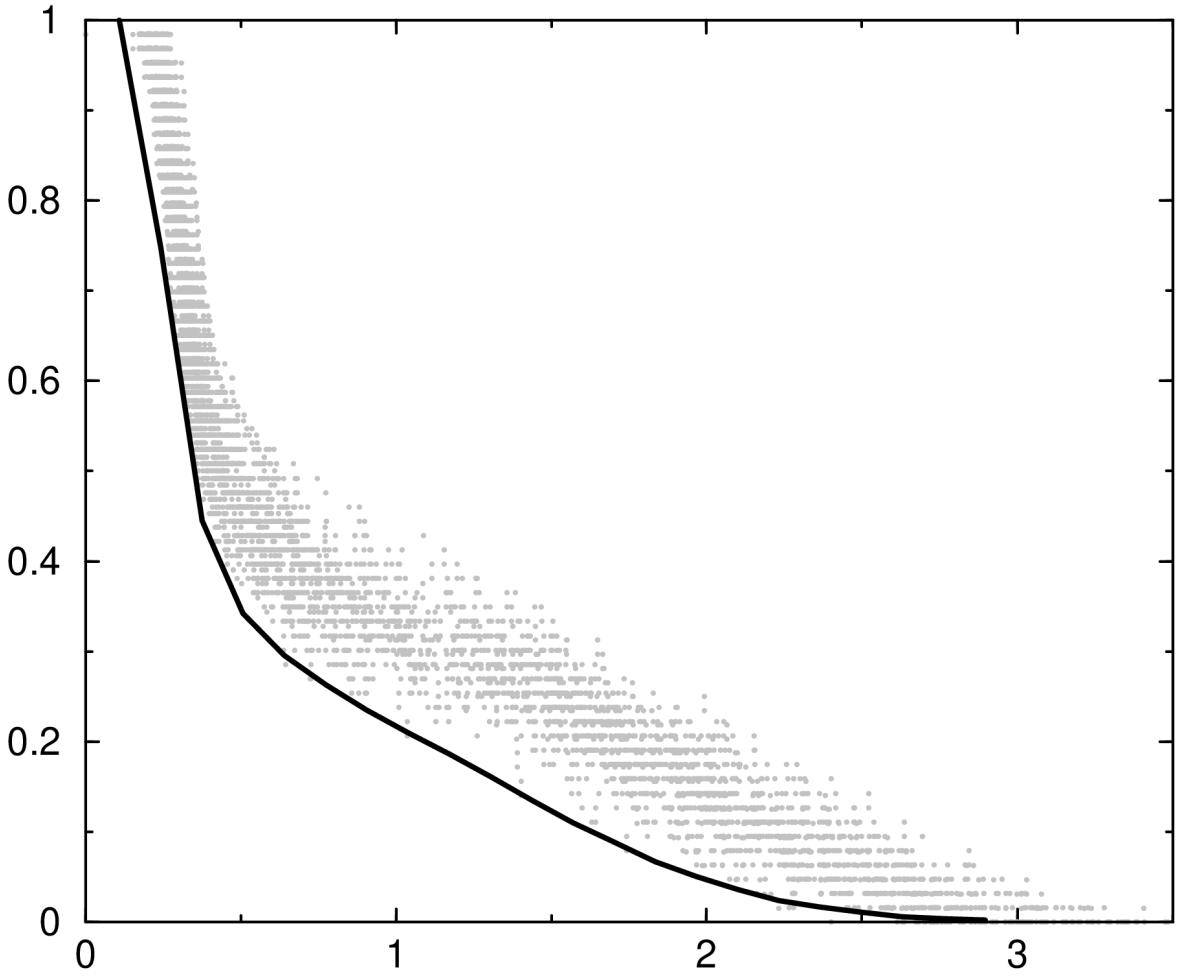

Fig.36 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 100K scatter plot (gray scatter plot) for the cln025 peptide structure.

The scatter plot appears to be higher (has a higher set of values) than the line and they do not overlap with each other.

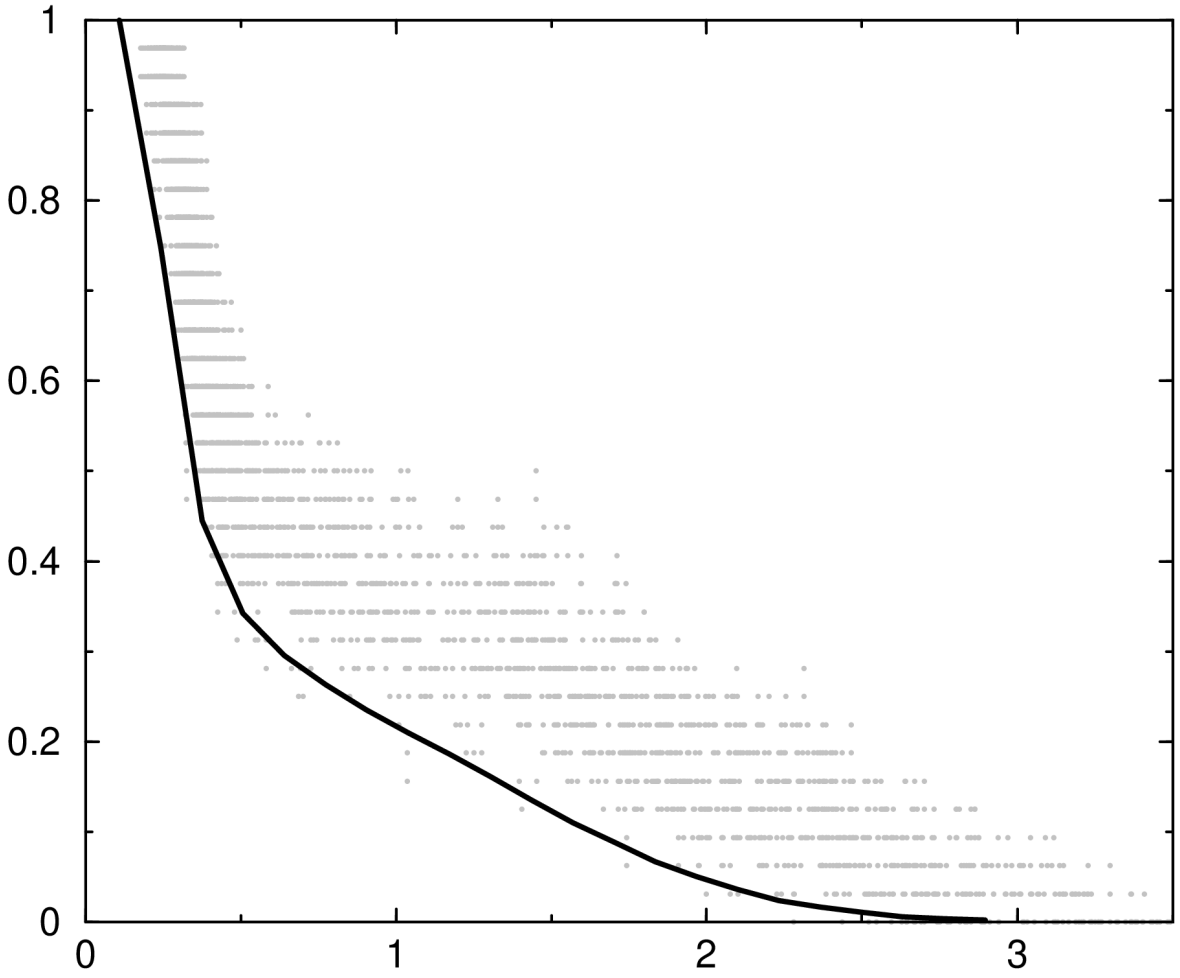

Fig.37 Comparison between the results obtained using a 10345x10345 matrix (black line) and a 200K scatter plot (gray scatter plot) for the cln025 peptide structure.

The scatter plot is spread in a big area of the graph, it is higher (has a higher set of values) than the line and the two (the scatter plot and the black line) do not overlap almost anywhere.

2n0x:

In the following graphs are presented the results that derive from the "Good-Turing" method (black lines) and the "independent" method (scatter plots) for the 2n0x peptide structure.

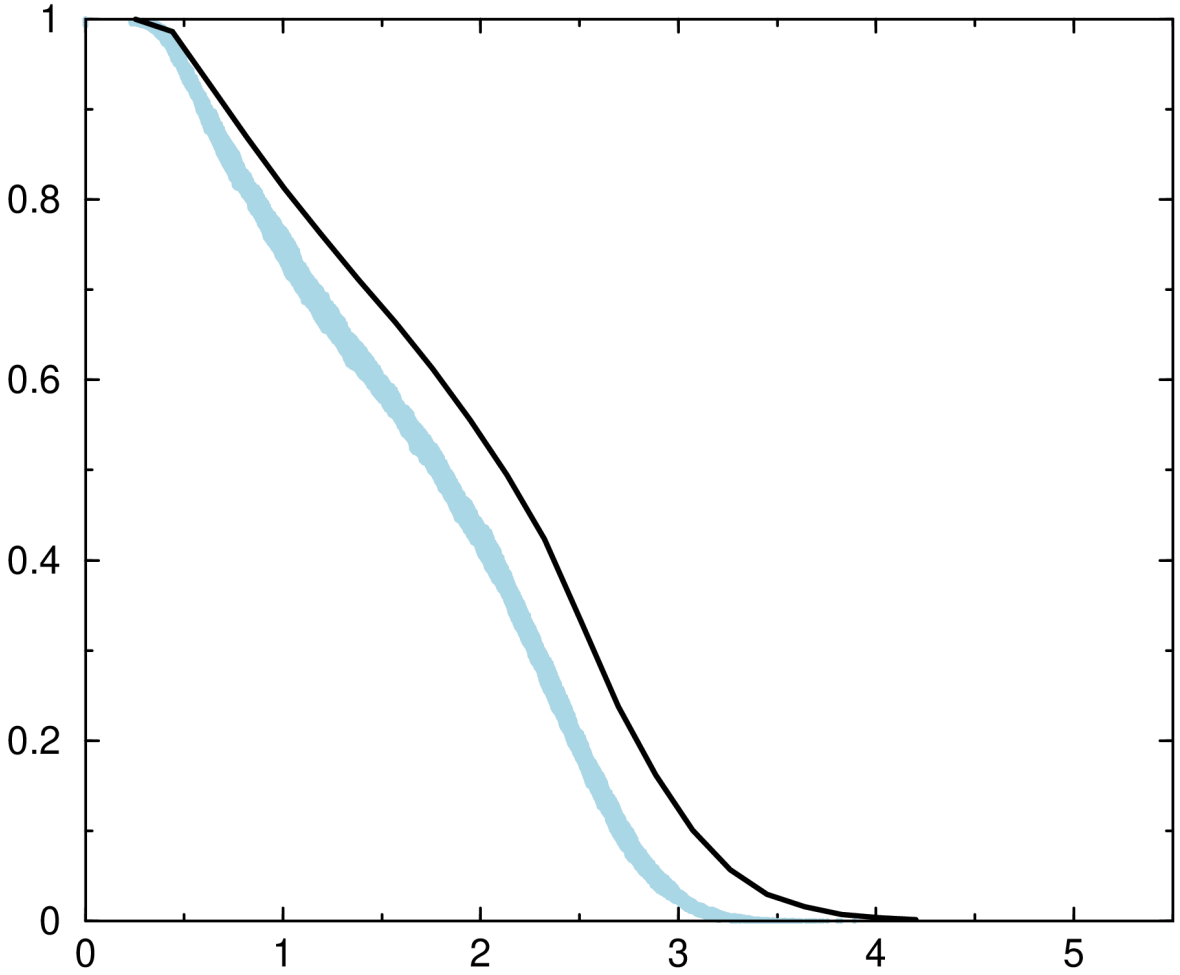

Fig.38 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 10K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The scatter plot-which is the result of the "independent" method-is close to the black line, but they do not overlap in almost any area of the graph.

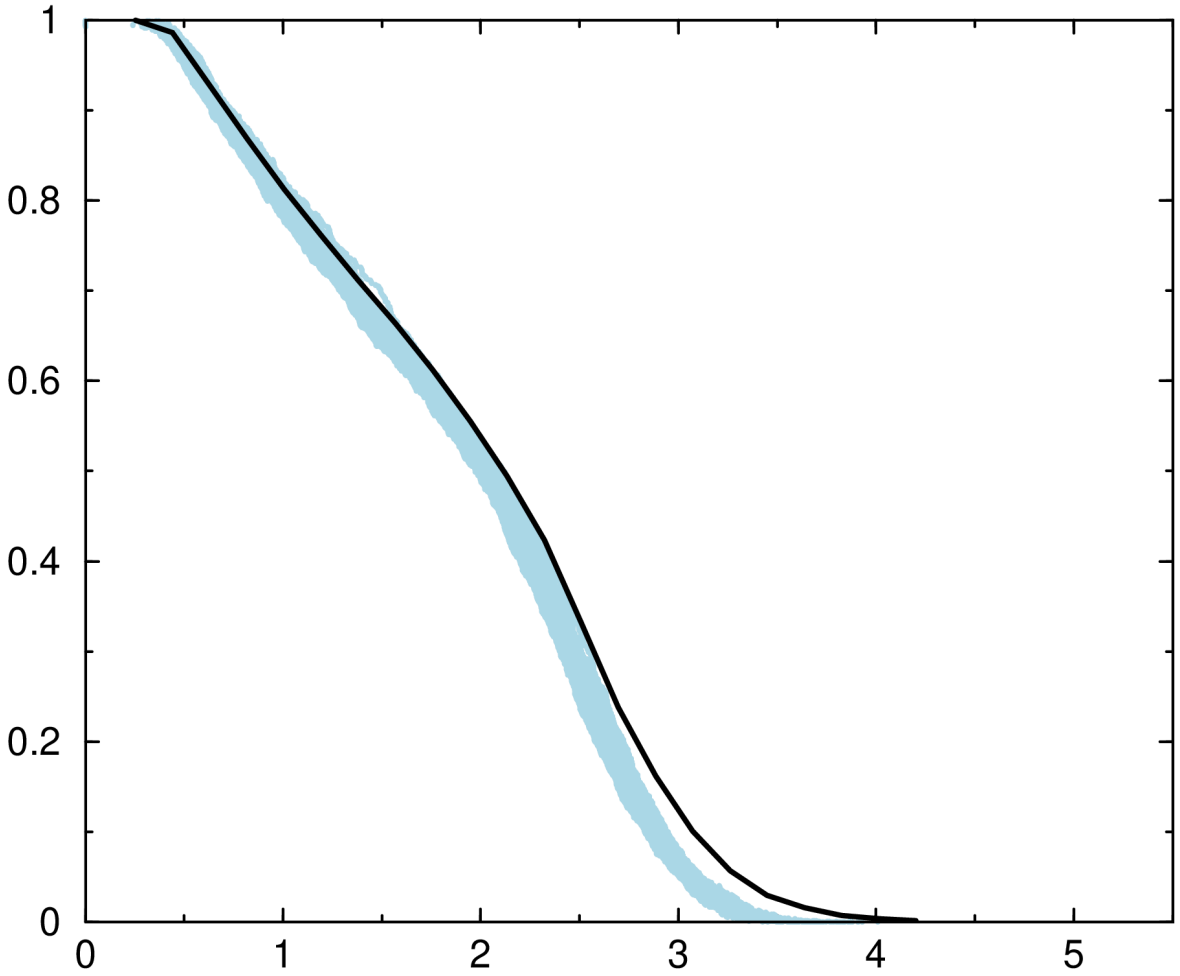

Fig.39 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 20K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

It is obvious that the 10K increase in the step has resulted in a better scatter plot than the one in Fig.38. The line and the scatter plot overlap in a few areas of the graph, but the scatter plot is placed lower (has a lower set of values).

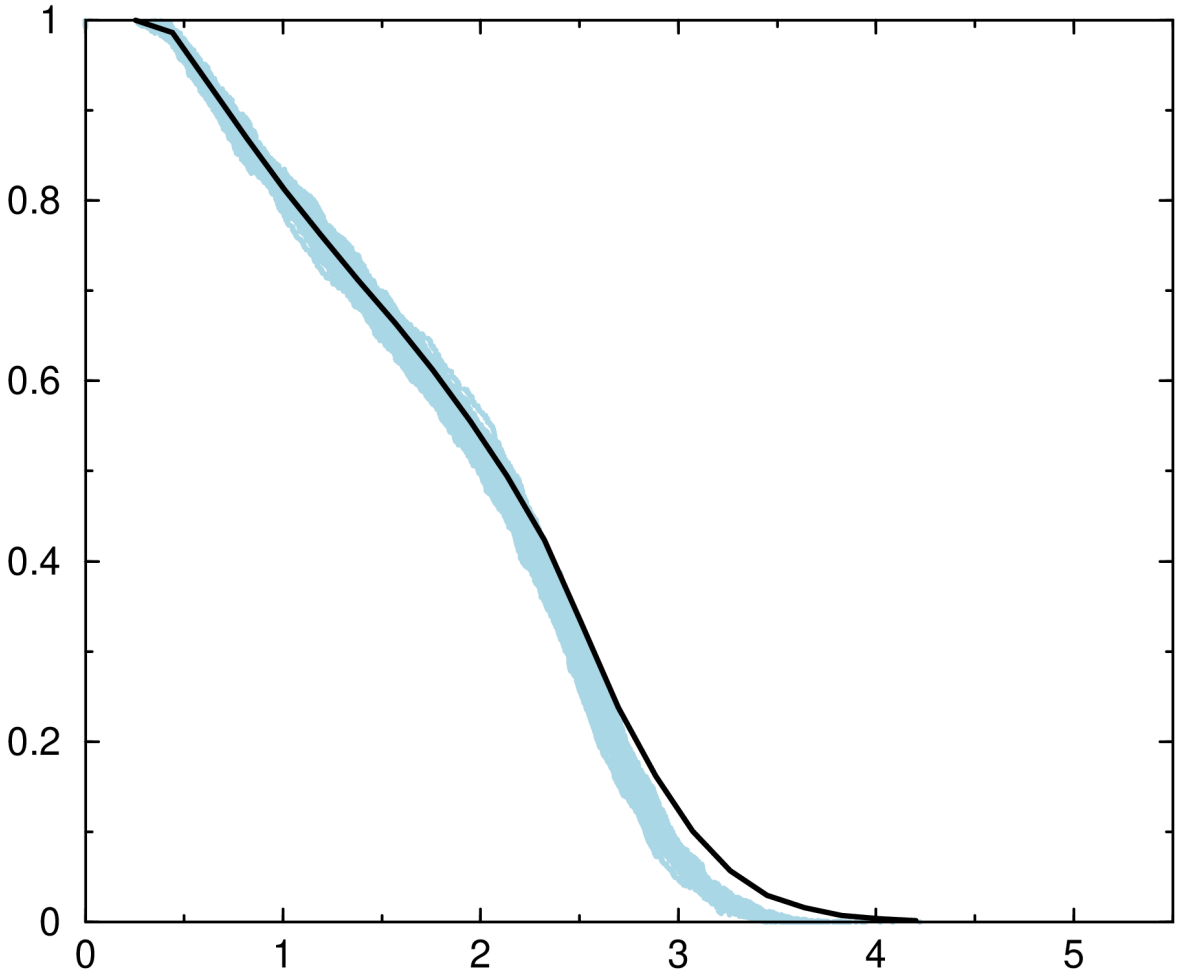

Fig.40 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 22K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The line and the scatter plot overlap in many areas of the graph and in almost all the areas where they overlap, the line appears to be “in the middle” of the scatter plot.

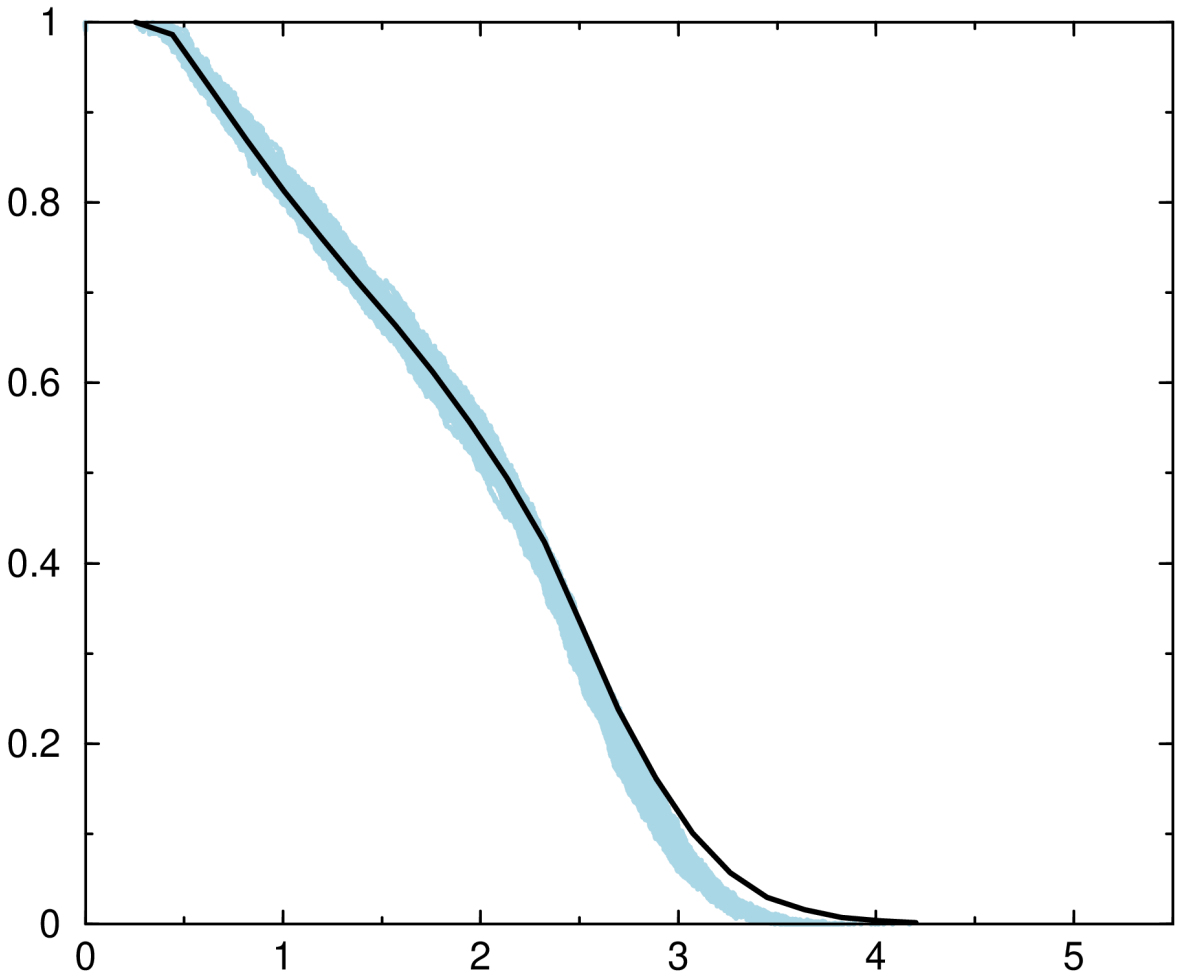

Fig.41 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 25K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

As the step increases, the line and the scatter plot overlap, but in a few areas the line is no longer “in the middle” of the scatter plot and in these areas, the scatter plot has a higher set of values than the line. Compared to Fig.40, it is important to note that in this graph (Fig.41) the line although not “in the middle”, is much closer to the scatter plot for the 2-4 Å RMSD values.

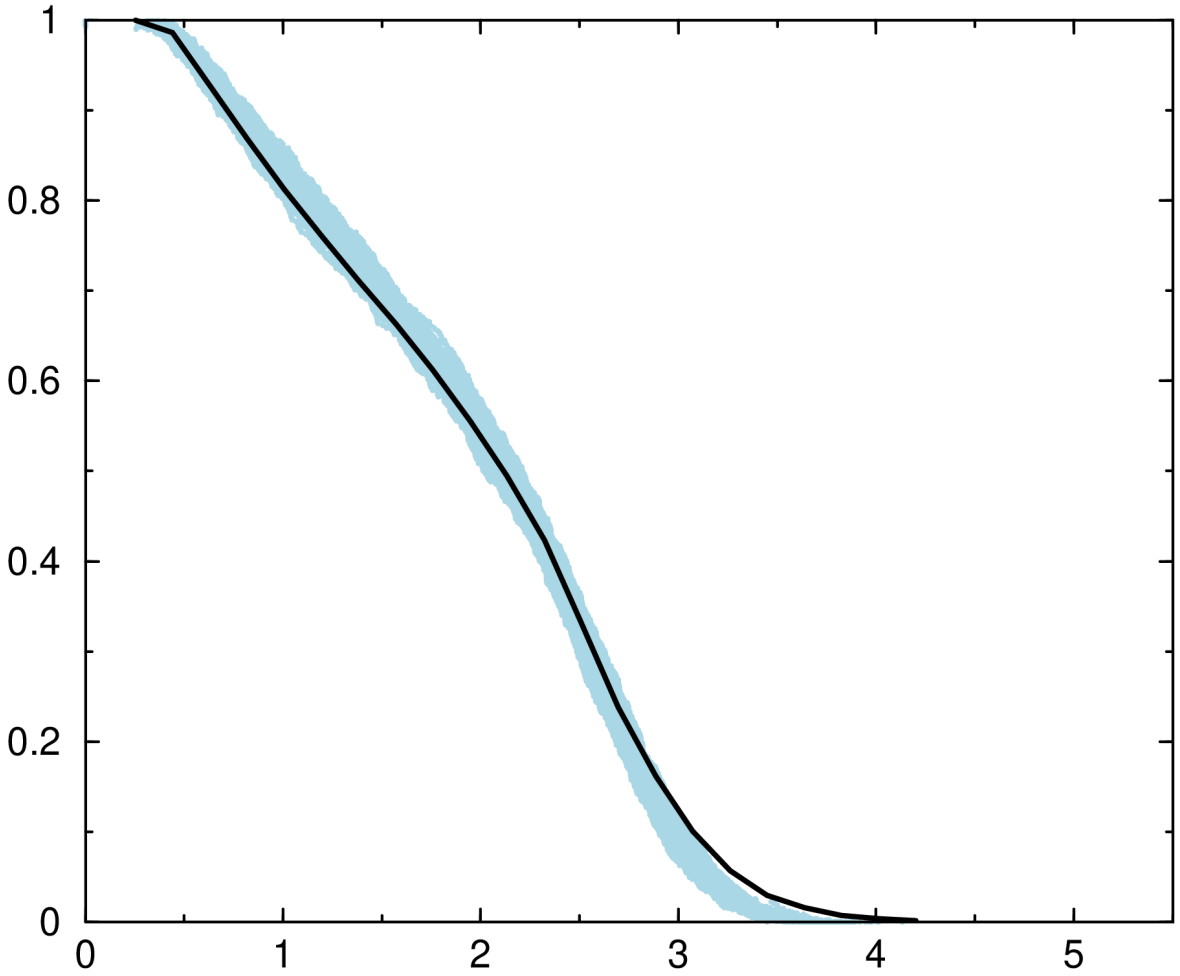

Fig.42 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 28K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The line overlaps with the scatter plot in almost every area of the graph, which is the case for Fig.40 and Fig.41 as well, but the scatter plot is located higher (has a higher set of values) and in some of these areas the line appears to be “in the middle” of the scatter plot (in the areas approximately between the 2-2.5 Å RMSD values).

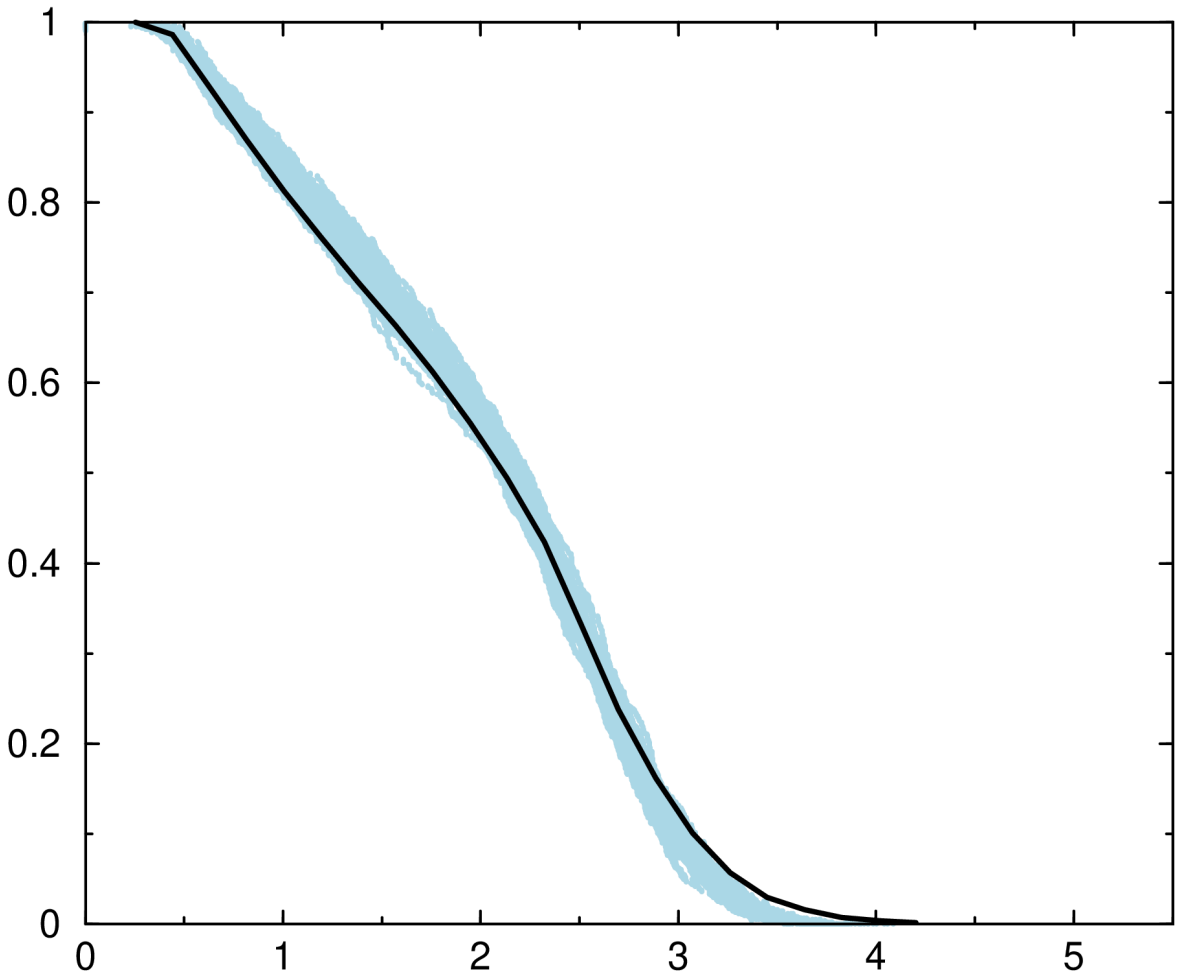

Fig.43 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 30K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The line overlaps with the scatter plot in almost every area of the graph, which is the case for Fig.40-42 as well, but in this graph the scatter plot is located higher (has a higher set of values).

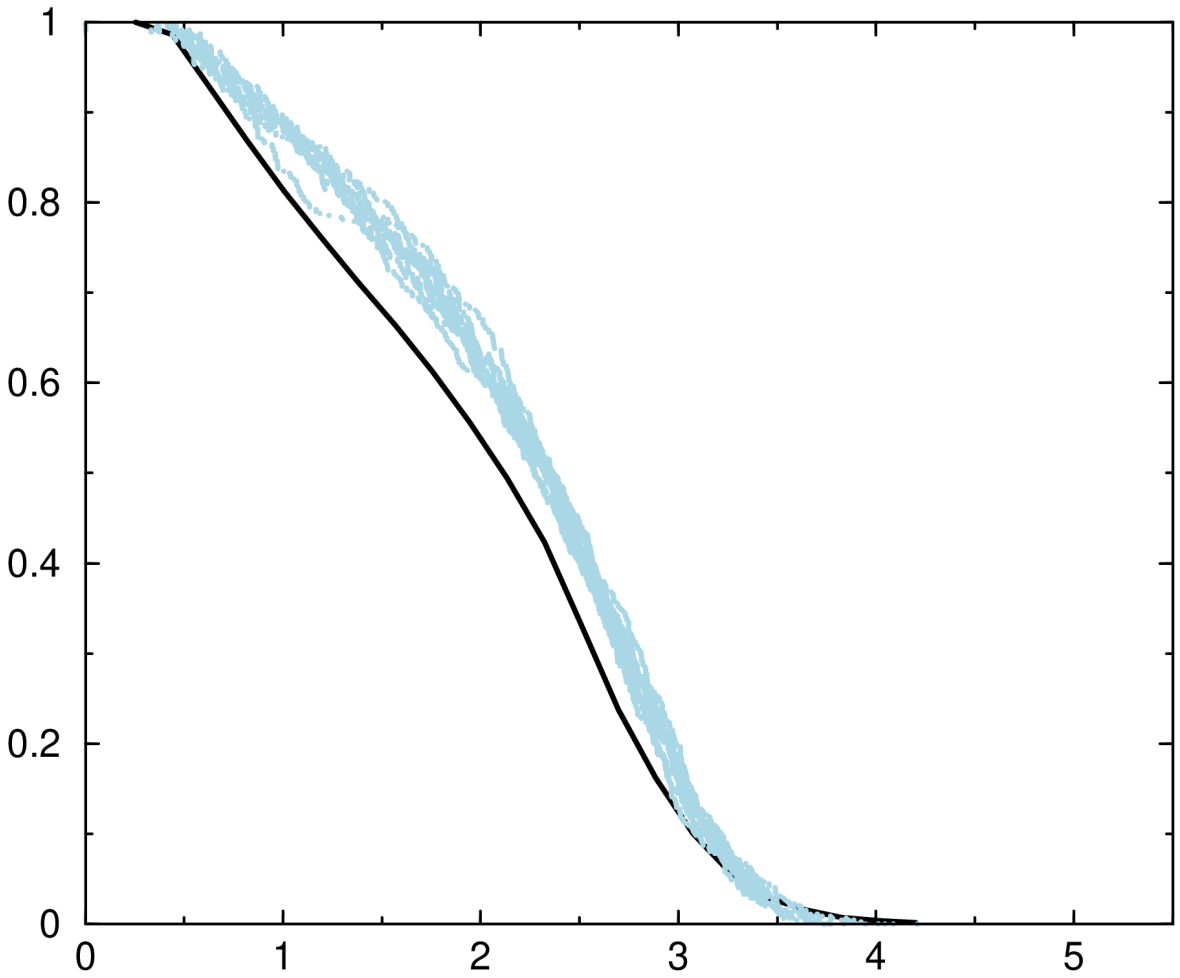

Fig.44 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 50K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The scatter plot is higher (has a higher set of values) than the line and the two (the scatter plot and the black line) overlap mainly at the areas for the bigger RMSD values (3-4.5 Å).

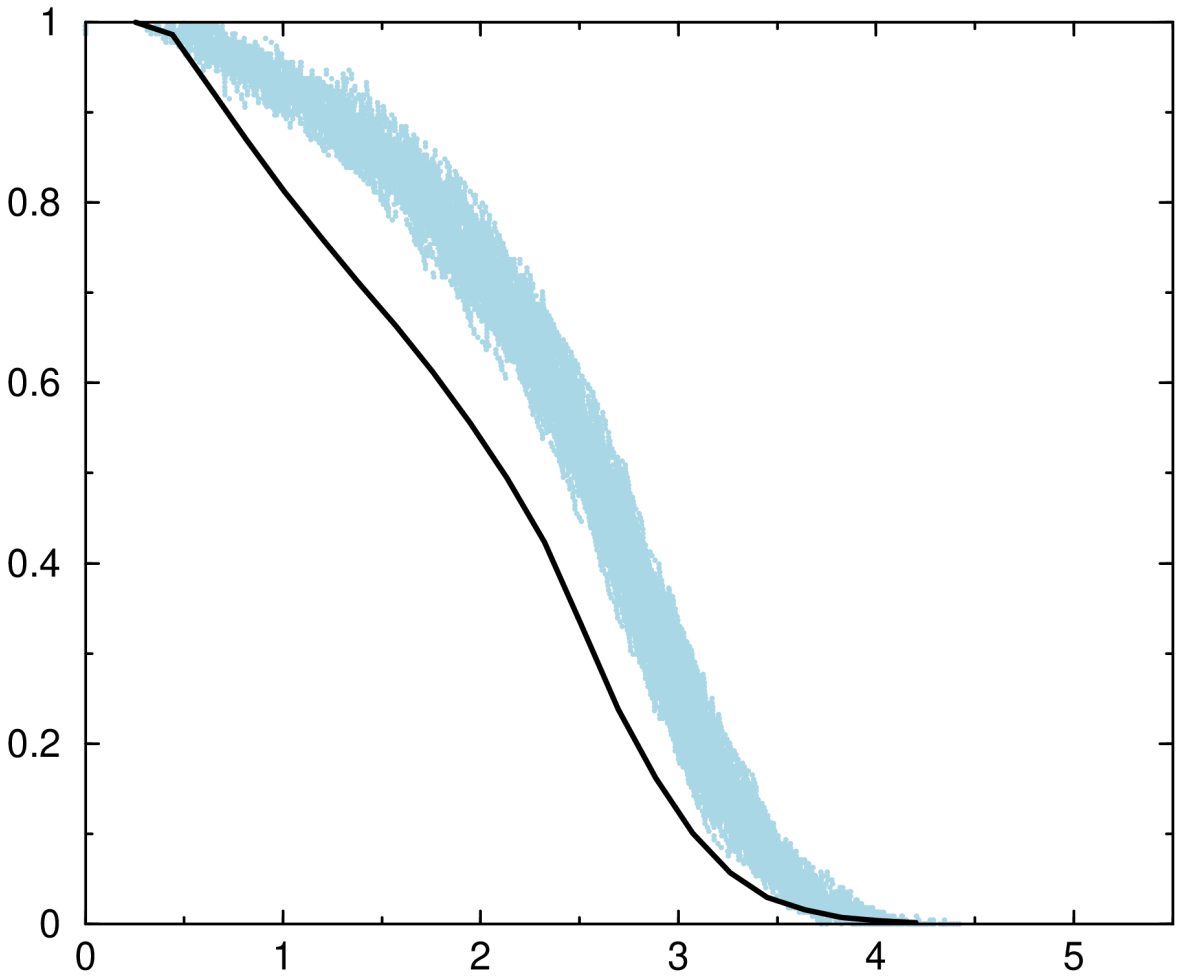

Fig.45 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 100K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The scatter plot is higher (has a higher set of values) than the line and the two (the scatter plot and the black line) do not overlap almost anywhere.

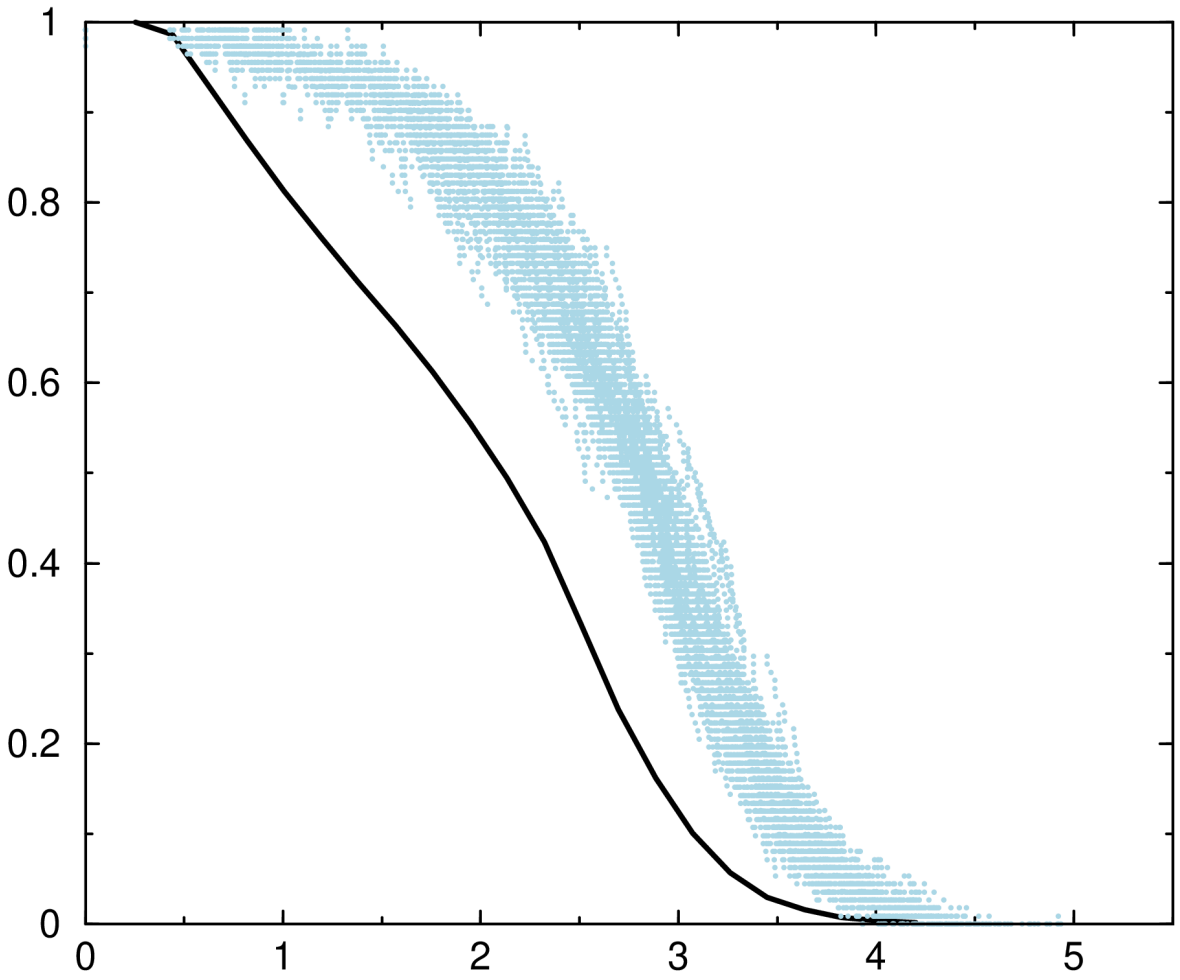

Fig.46 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 200K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The scatter plot is spread in a big area of the graph, it is higher (has a higher set of values) than the line and the two (the scatter plot and the black line) do not overlap almost anywhere.

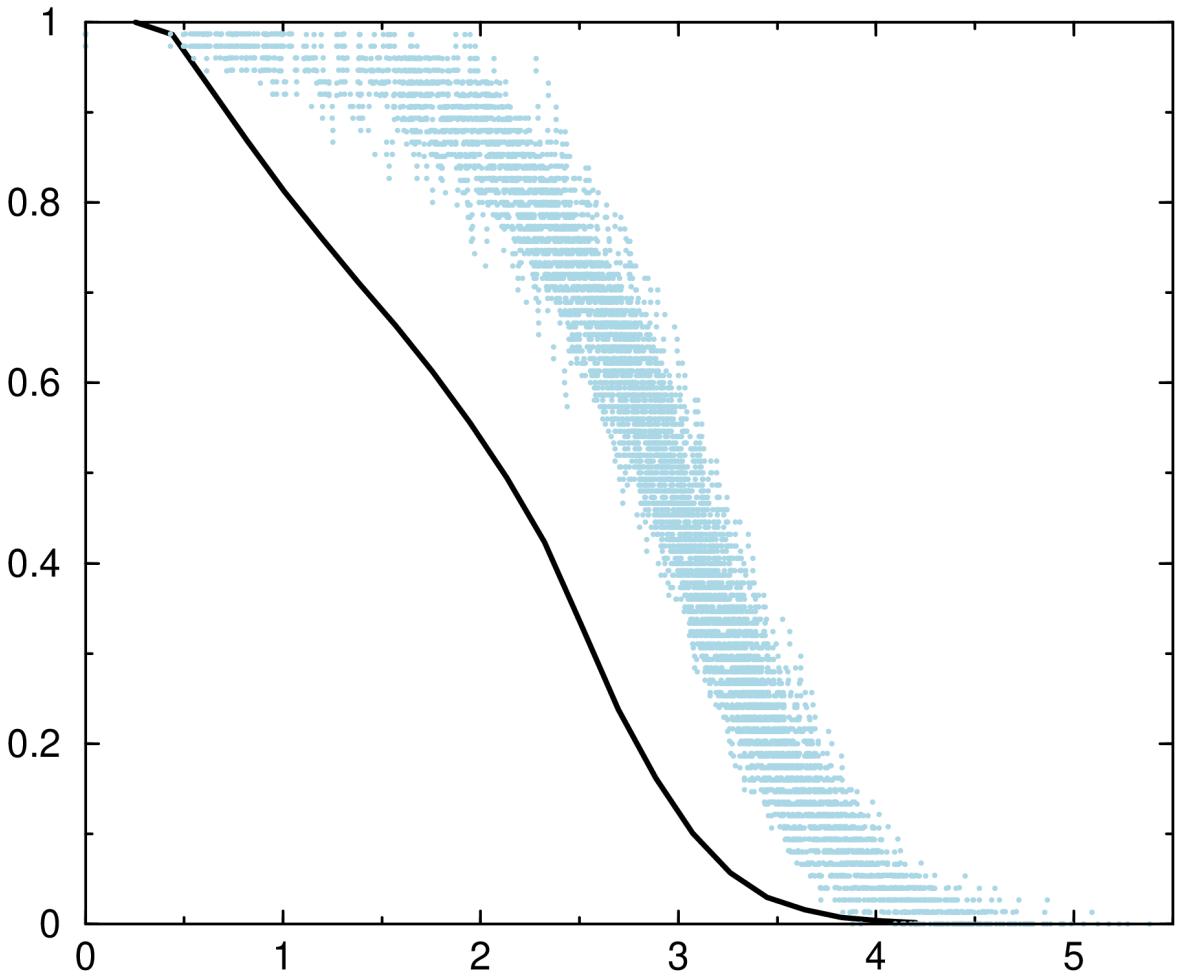

Fig.47 Comparison between the results obtained using a 10264x10264 matrix (black line) and a 300K scatter plot (light blue scatter plot) for the 2n0x peptide structure.

The scatter plot is “spread” in a big area of the graph, it is higher (has a higher set of values) than the line and the two (the scatter plot and the black line) do not overlap almost anywhere (in a similar fashion as in Fig.45 and Fig.46).

3.3 The “Jude” method

For the “Jude” method many different trajectories for proteins or peptide structures were used to test the quality of the method. The samples showcase a variety of specific properties (pertaining to the size, folding rate and behavior), which is desirable, considering it makes evident whether the method is adequate or not, for any kind of protein or peptide structure. More trajectories were used as samples for the evaluation of the “Jude” method, than the ones presented in this section. The following results successfully highlight the way the method works for different kinds of proteins or peptide structures and the results that are not included, do not lead to other conclusions, regarding the efficacy and the reliability of the method.

cln-ILDN:

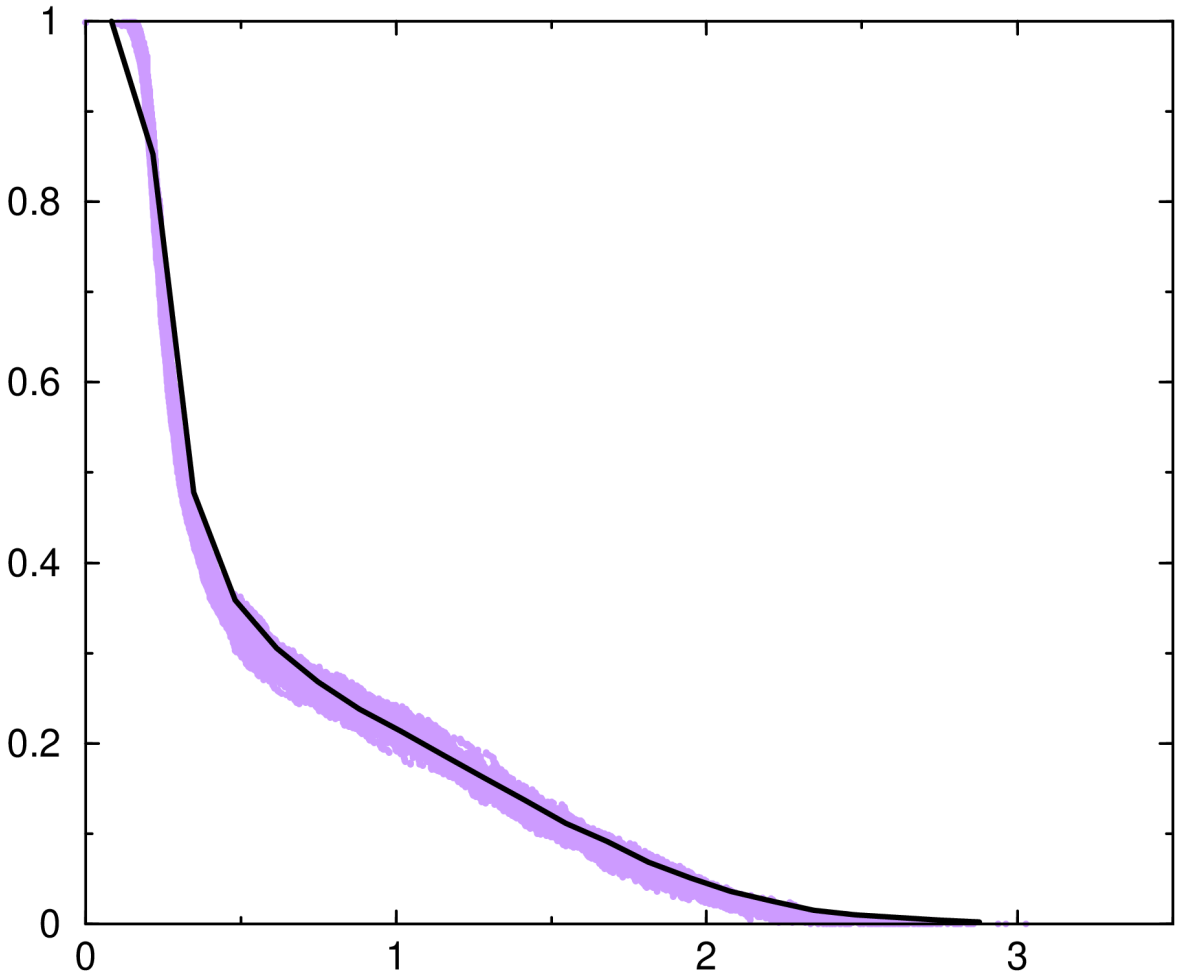

Fig.48 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot overlap in almost every possible area. In the majority of the areas where the two overlap, the scatter plot is found to have a slightly lower set of values than the line. The line appears to be "in the middle" of the scatter plot in the areas approximately between the 0.75-1.25Å RMSD value.

A31P:

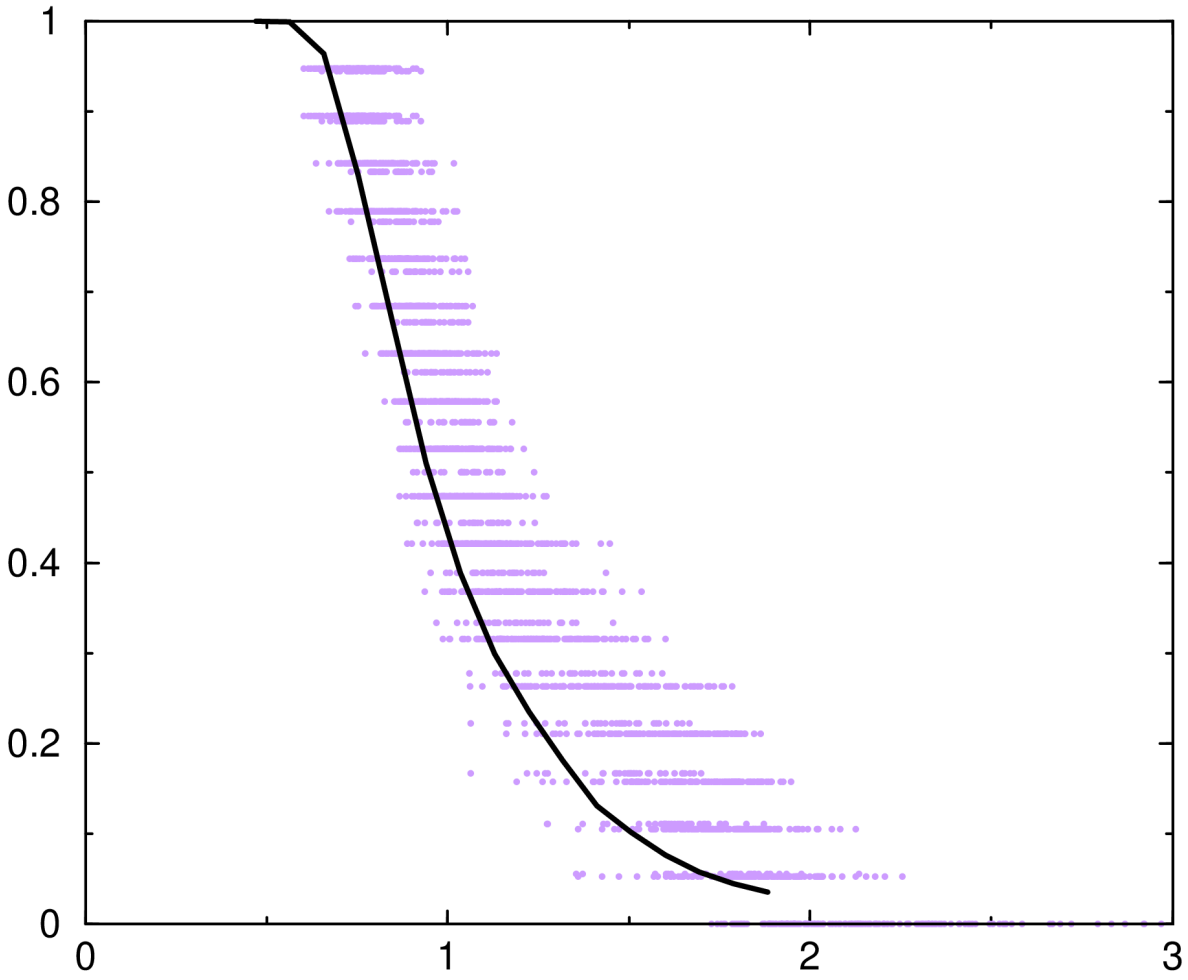

Fig.49 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot overlap with each other. The scatter plot is "spread" across the graph, showcasing a big range in the probability of unobserved configurations for different RMSD values. The line has a lower set of values than the scatter plot, but never reaches a possibility of 0 for unobserved configurations at any RMSD value.

2n0x:

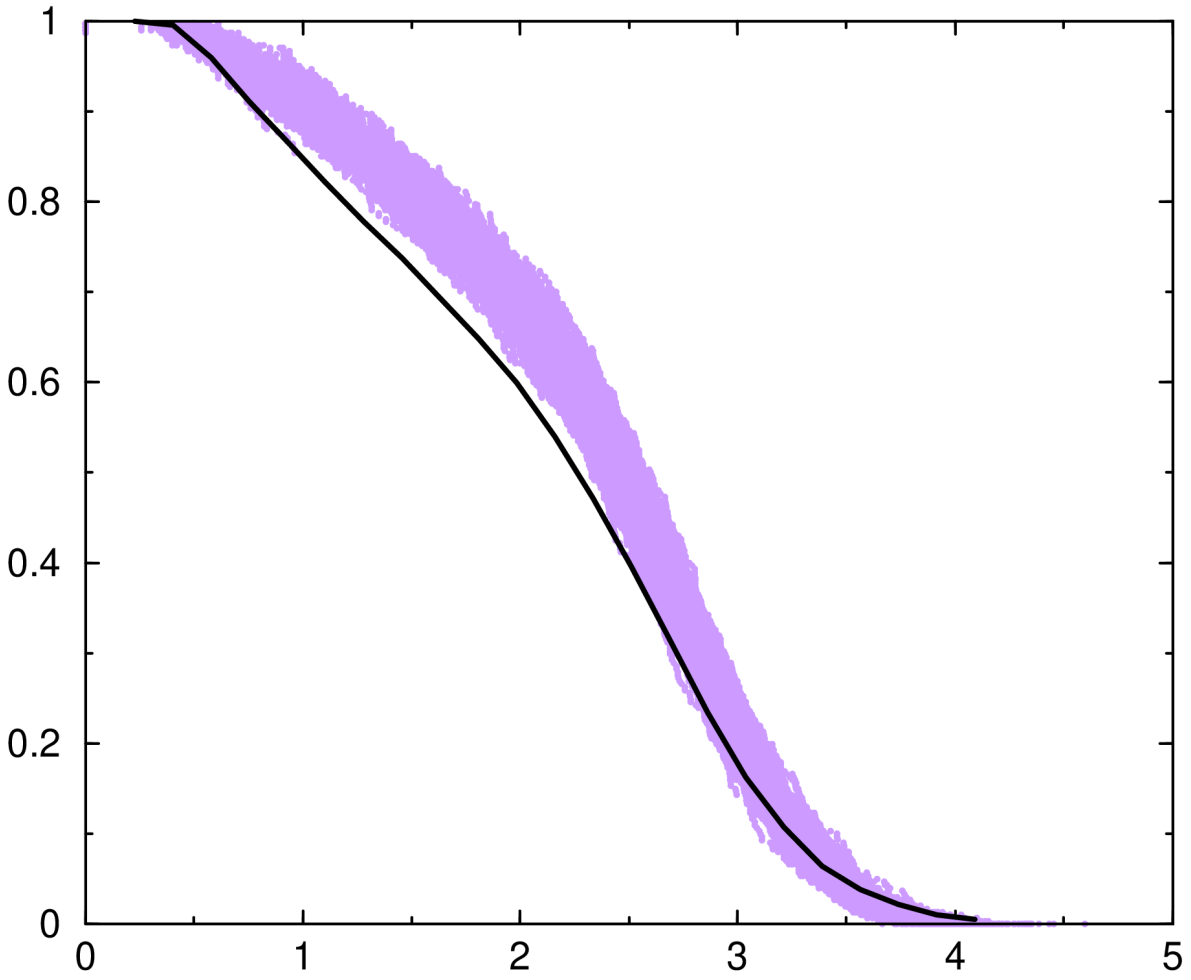

Fig.50 Comparison between the line that occurs from the classic “Good-Turing” method (black line) and the scatter plot that derives from the application of the “Jude” method (light violet).

The scatter plot and the line overlap in some areas and in the ones where they do not, they are relatively close to each other. The scatter plot also has a higher set of values than the line.

6NM2:

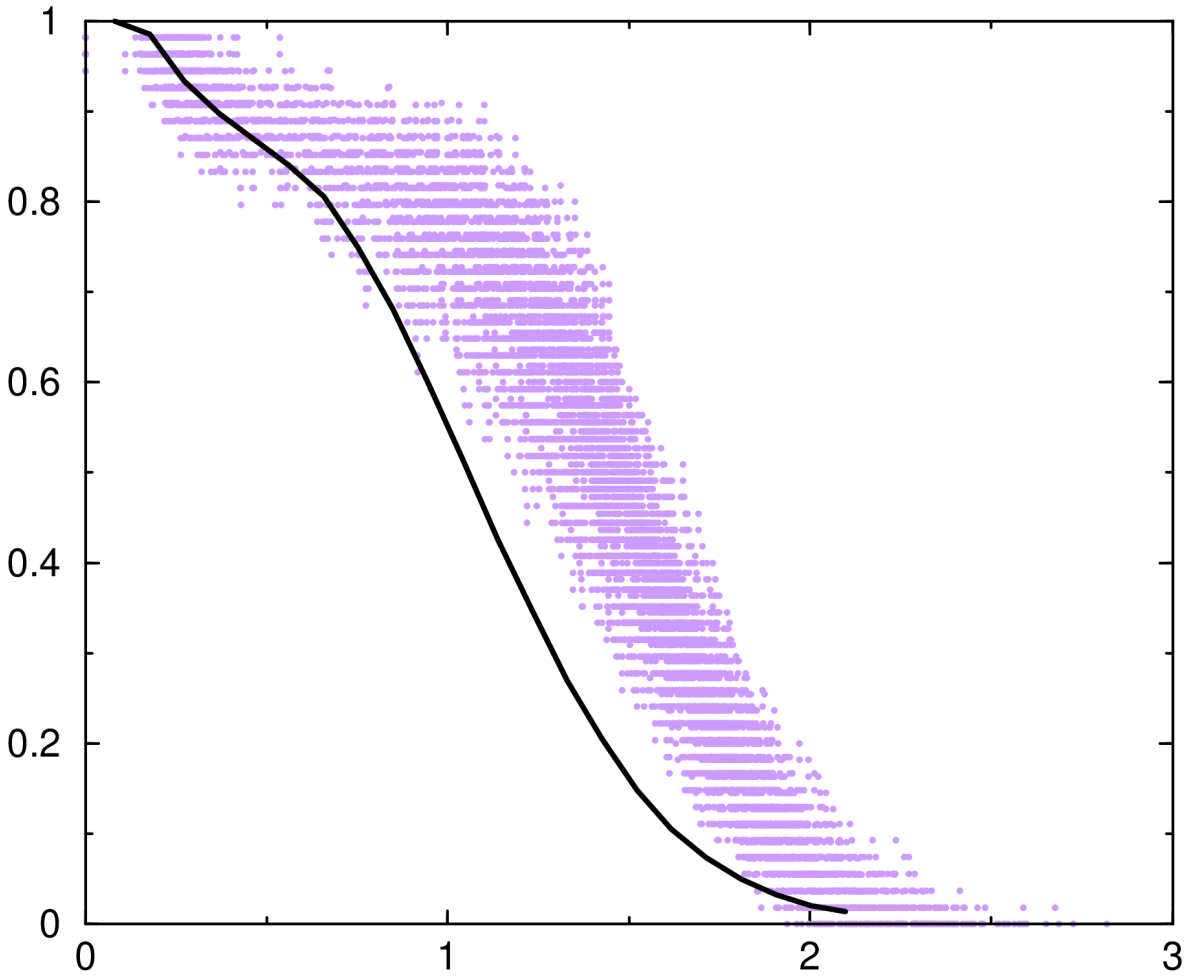

Fig.51 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot overlap in very few areas, with the scatter plot showcasing quite a big range in the probability of unobserved conformations for different RMSD values, with a higher set of values than the line. The line on the other hand, never reaches the possibility of 0 unobserved conformations at any RMSD value.

Nat-STAR-notails:

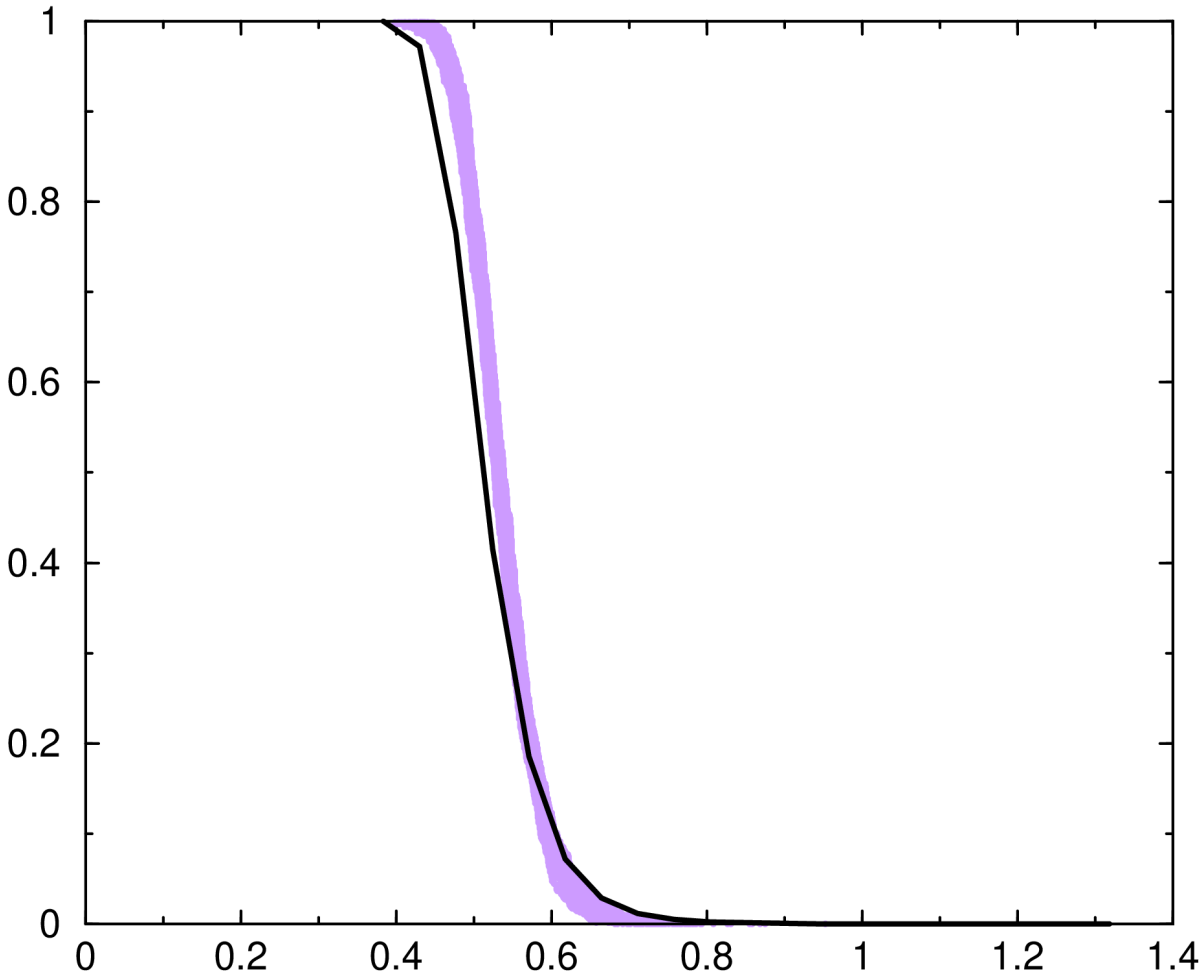

Fig.52 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot overlap in some areas and in the areas where they do not, they are quite close with each other. The scatter plot is more in the "right" on the graph (has a higher set of values) than the line, with the exception of the area where the probability of unobserved configurations is 0 or near 0.

Nat-STAR:

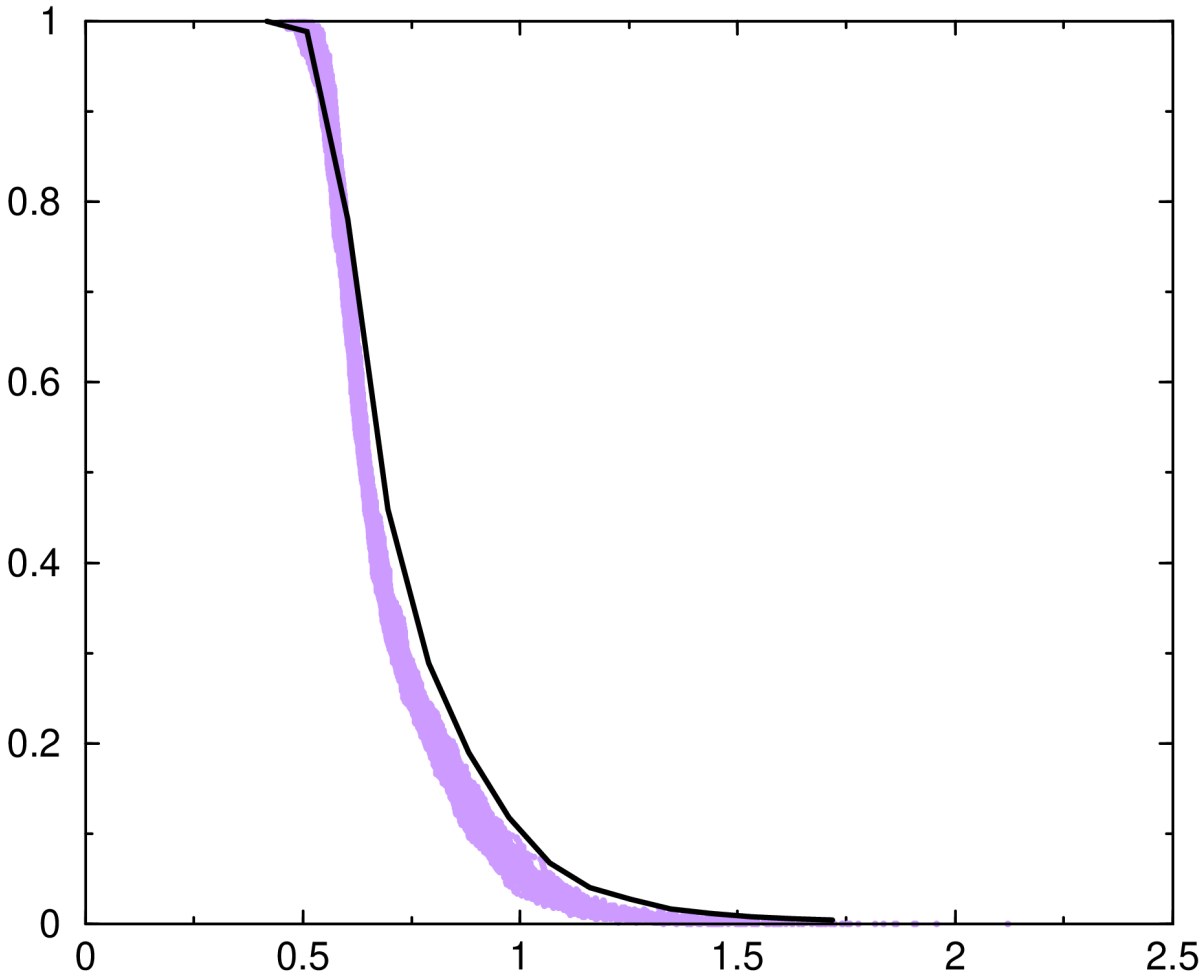

Fig.53 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot overlap in some areas of the graph and in the ones where they do not, they are quite close to each other. The scatter plot is more on the "left" of the graph (has a lower set of values) than the line.

Nat-A31P:

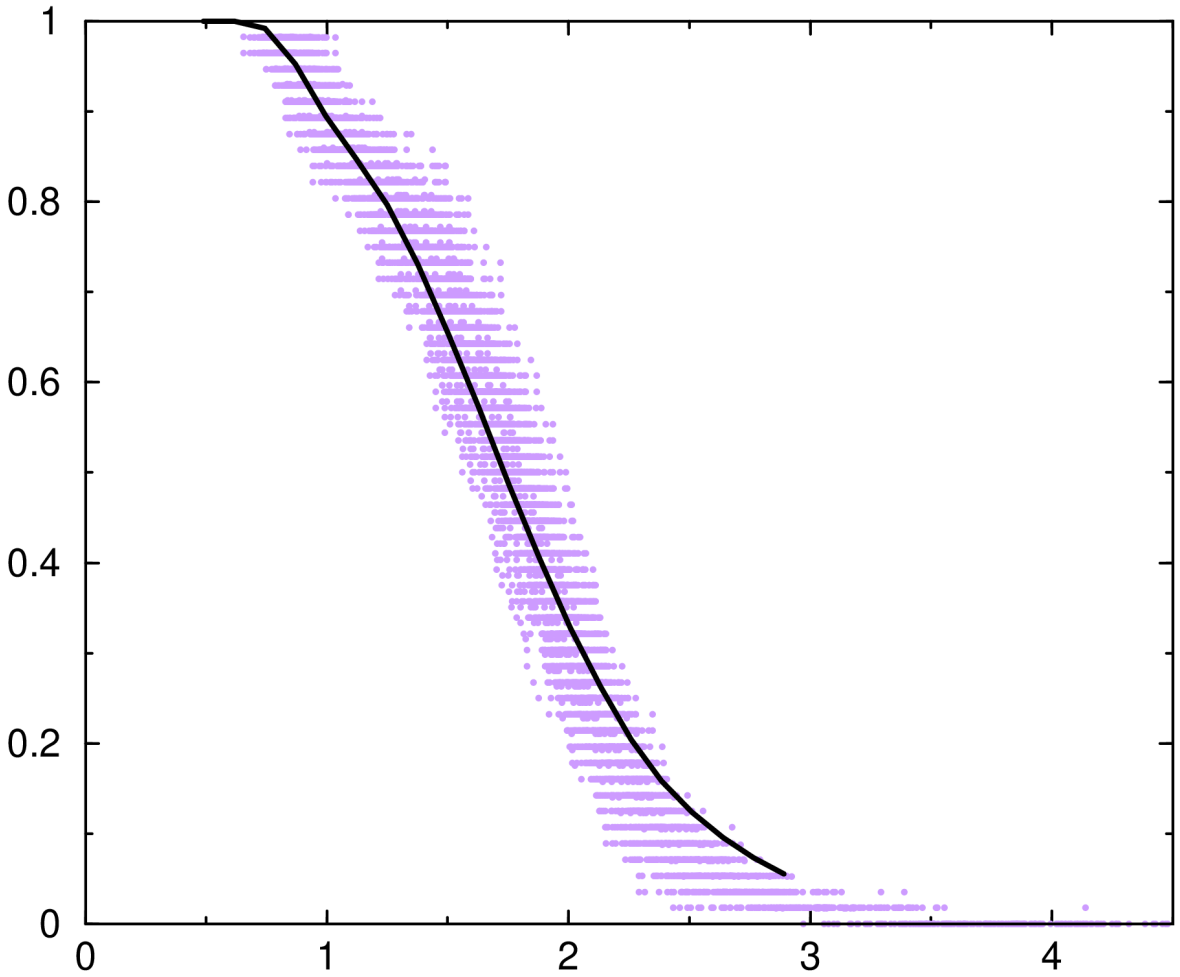

Fig.54 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line overlaps with the scatter plot in every area possible, with the scatter plot being more on the "right" than the line (having a higher set of values) for lower RMSD values and more on the "left" (having a lower set of values) for higher RMSD values. The line also never reaches the possibility of 0 unobserved configurations at any RMSD value.

pdb2mq2:

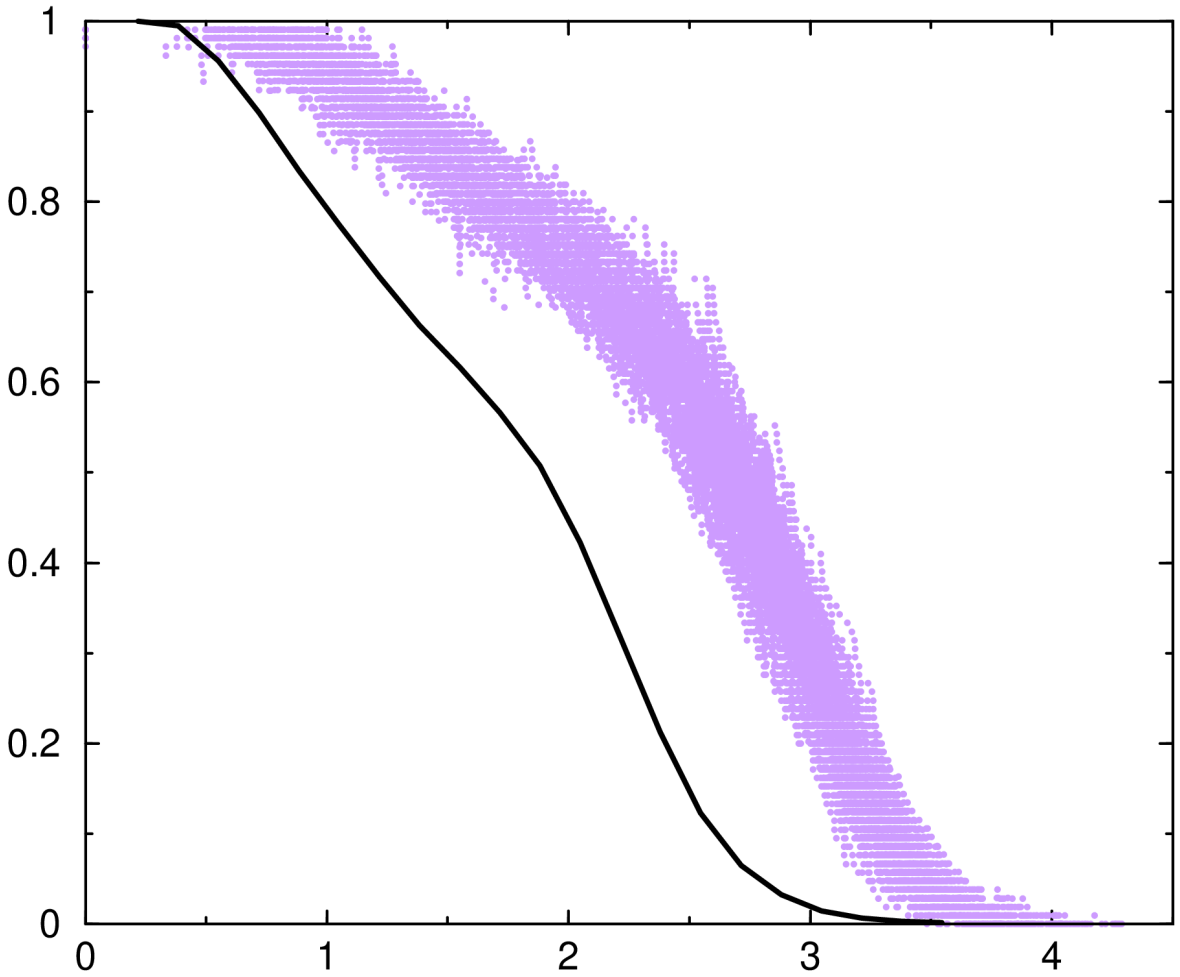

Fig.55 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot are not close to each other, with the only areas of slight overlap located at the very low RMSD and very high RMSD values. The scatter plot is more on the "right" (has a higher set of values) than the line and it has a big range of different pairs of data, which is responsible for how much the scatter plot "spreads" across the graph.

NFGAILS:

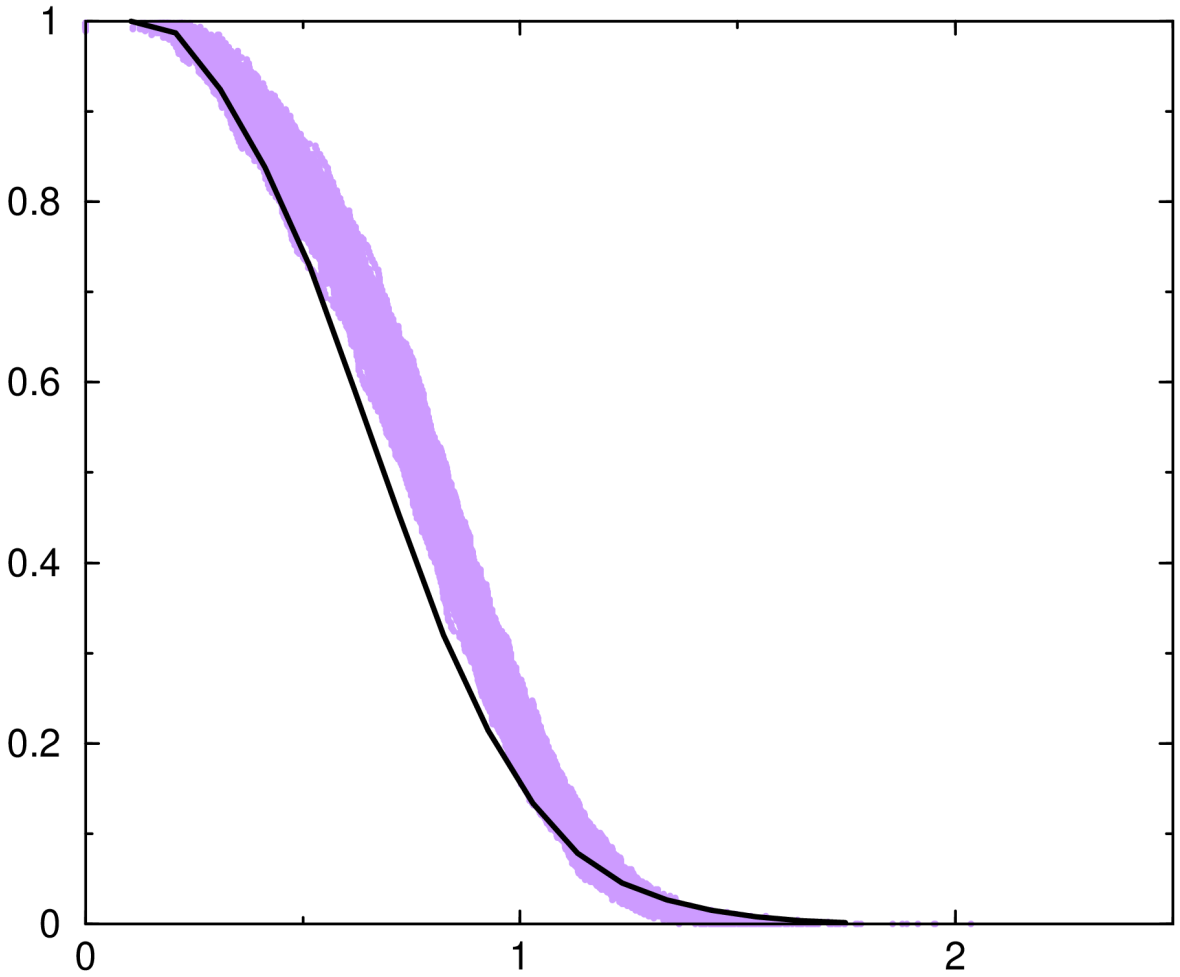

Fig.56 Comparison between the line that occurs from the classic “Good-Turing” method (black line) and the scatter plot that derives from the application of the “Jude” method (light violet).

The scatter plot is located more on the “right” (has a higher set of values) than the line. The line and the scatter plot overlap in a few areas and in those where they do not, they appear relatively close with each other.

SarsTM:

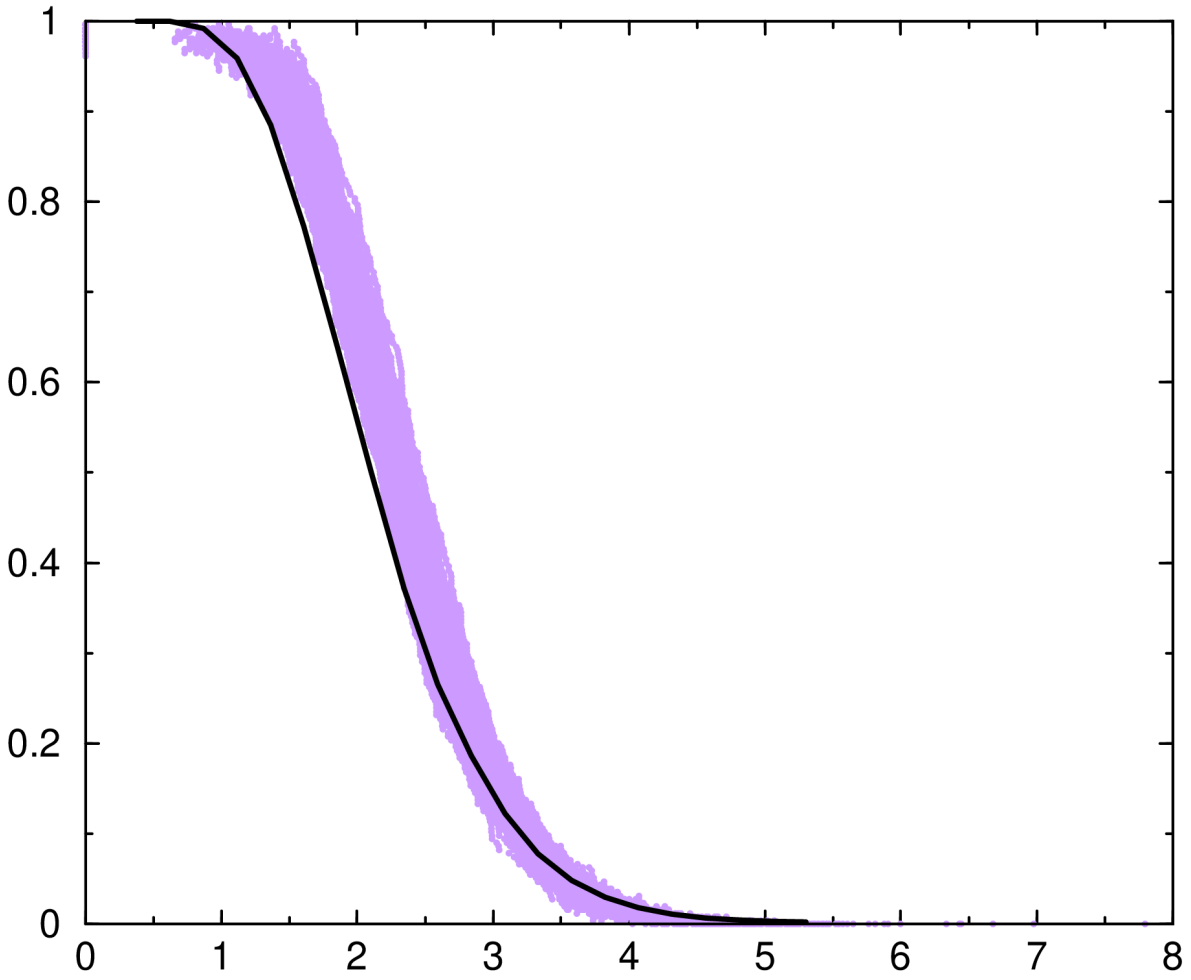

Fig.57 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot overlap in many areas of the graph and in those where they do not, they appear very close to each other. The scatter plot is located more on the "right" (has a higher set of values) on the graph. It is also important to mention, that between the RMSD values of 3-4Å, the line is "in the middle" of the scatter plot.

InflA:

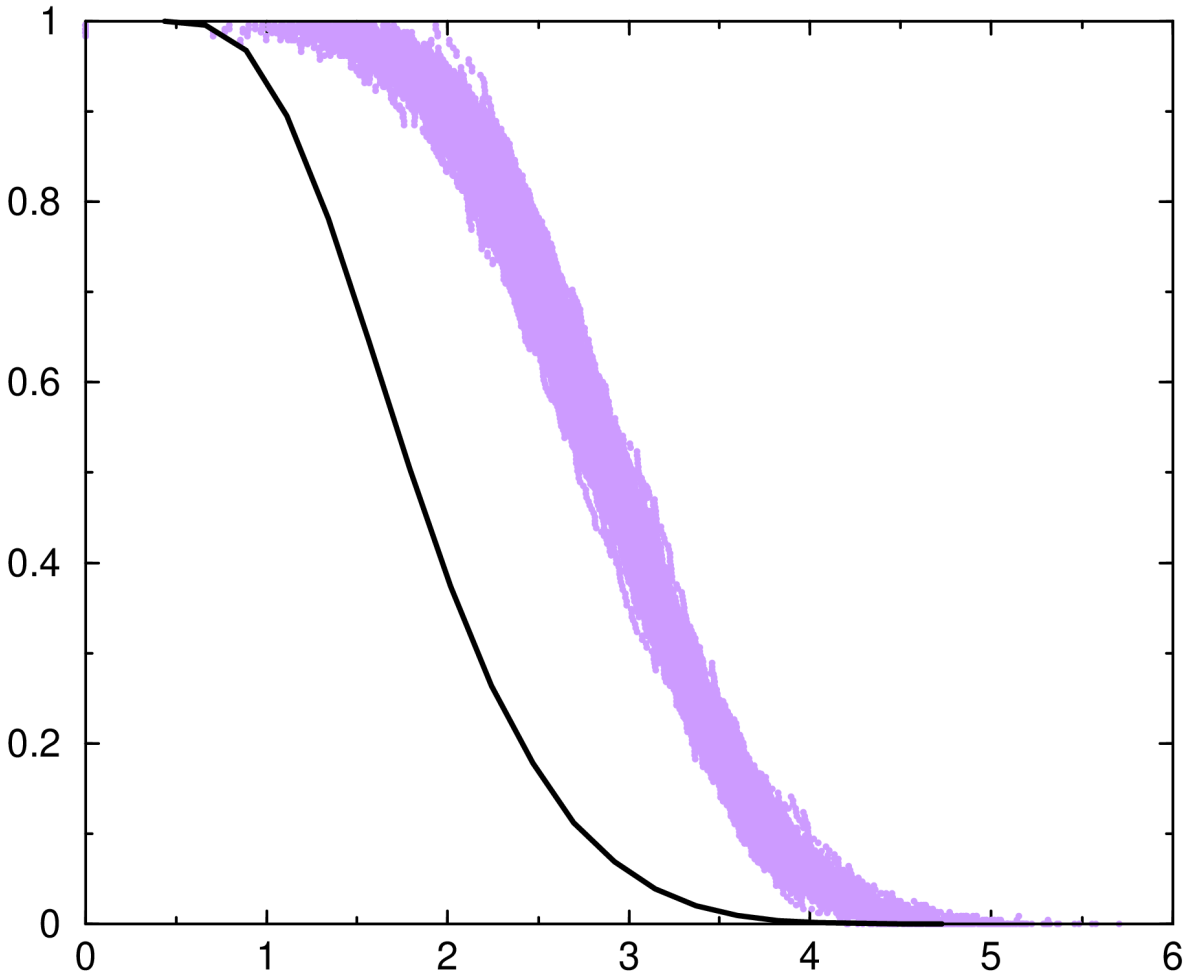

Fig.58 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot are not close to each other overall and the only areas where they overlap are located at high RMSD values, or at very low RMSD values (practically the only area actually worth mentioning is between the 4-5Å area). Compared to the line, the scatter plot is more on the "right" side (has a higher set of values) of the graph.

5glh:

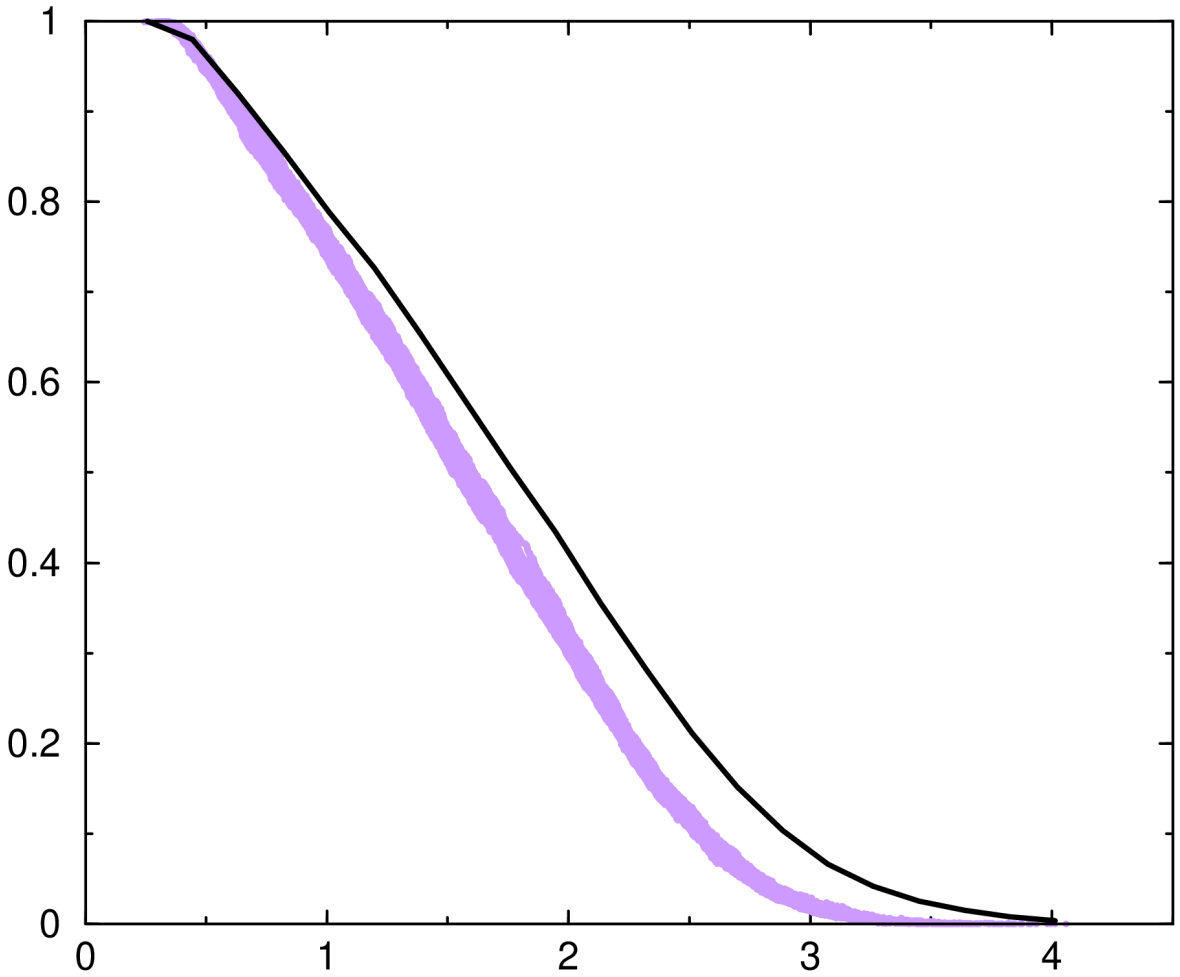

Fig.59 Comparison between the line that occurs from the classic "Good-Turing" method (black line) and the scatter plot that derives from the application of the "Jude" method (light violet).

The line and the scatter plot are mainly relatively close to each other, with a single area of overlap spotted at low RMSD values. Compared to the line, the scatter plot is more on the "left" (has a lower set of values) on the graph.

4. Discussion

4.1 The classic “Good-Turing” method and the “max of mins” method

The results pertaining to the classic “Good-Turing” method are as expected; when the step (stride) between frames–which results in different sized matrices–is big, a shift towards the right can be observed on the graphs. When comparing the different methods (classic “Good-Turing” method and “max of mins” method), a shift towards the left on the graphs can be observed (see Fig.15 and Fig.25). Of course, a shift either towards the right or the left on the graphs was expected and it is understandable, since less data have been used to determine convergence, but the shift towards the left indicates that the “max of mins” method is more “optimistic” than the classic “Good-Turing” method (which calculates the whole matrix), indicating that convergence is reached earlier. A promising feature is the fact that the differences in the datasets of the “max of mins” method and the classic “good-turing” method are small. If one is to compare the proximity of the lines from each of the two methods (classic “Good-Turing” and “max of mins” method), when the trajectory of cln025 is sampled (Fig.15) and when the trajectory of 2n0x is sampled (Fig.25) the difference is obvious, with the lines deriving from the analysis of the trajectory of cln025 being very close to each other. This difference is easily attributed to the fact that 2n0x is a significantly more flexible peptide structure that reaches convergence later than cln025. Naturally, the classic “Good-Turing” method always provides more accurate results, since it utilises the whole matrix. With that being said, the importance of the fact that every dataset that derived from the different runs, using the “max of mins” method, is shifting towards the left side should be noted, because it means that this method is consistently a bit more “optimistic”, indicating better convergence than what is actually the case. An inconsistency in the results, that should also be mentioned, is the

sub-sampling factor, which does not gradually get smaller in every case (see Table 3 and Table 5), but there are some inconsistencies due to noise, although that fact does not cause any real concern, since even with that noise the results are still reliable.

4.2 The “independent” method

The results from this method show that the “independent” method is not a reliable enough method and without the results from the previous grcarma runs (classic “Good-Turing” and “max of mins” method), where the ideal and actual step of the program used is known, many runs of the "independent" method were required in order to determine the ideal step. Out of all the steps that have been opted in the “independent” method the ideal one appears to be between 11,5-12K, which is approximately two times (2x) the actual step that the program (grcarma) used (6100) to produce the 10345x10345 matrix for the cln025 peptide structure. From the comparison of the results deriving from the classic “Good-Turing” method, with the results deriving from the “independent” method, for the 2n0x peptide structure, it is obvious that the ideal step for the “independent” method cannot be determined as easily as it was for cln025 (a significantly more stable protein), since some steps are more in agreement with the data deriving from the classic “Good-Turing” method for certain values and other steps are more in agreement with the data deriving from the classic “Good-Turing” method for certain other values of equal importance. Of course, the inability to have an ideal step for the “independent” method with the 2n0x peptide structure as a sample, can be attributed to its instability but it is surely not encouraging, further indicating the need for the step (stride) between frames to be determined based on a more concrete method. Another thing that should be clarified, is the fact that other matrices apart from the 10264x10264 matrix,

the 29834x29834 and the 15017x15017 matrices, were used as reference, in order to evaluate the efficiency and reliability of the “independent” method, but since the conclusions from these comparisons with the classic “Good-Turing” method, are such that they do not make the “independent” method appear more or less reliable, these results are not included. Experiments with even more steps (strides) between frames were conducted as well (for the assessment of the “independent” method), but again it would be excessive to include them after the method turned out to be unsatisfactory.

4.3 The “Jude” method

The trajectories tested indicate that the “Jude” method is mainly more “pessimistic” (i.e. a shift towards the right on the graphs can be observed for the scatter plot) (~67%), compared to the classic ‘Good-Turing” method. It is evident that the “Jude” method produces better quality results, when the protein sequence is such that the peptide structure is stable. Additionally, in 75% of the samples, the scatter plot and the line are not overlapping in every area possible, let alone have the line “in the middle” of the scatter plot. The fact that the method is mainly “pessimistic” is not disheartening, because it is better that this less accurate method does not assume convergence has been reached earlier than when it actually does. However, the ideal scenario would have the line–which is the product of the classic “Good-Turing” method–“in the middle” of the “Jude” method’s scatter plot. The “Jude” method cannot be characterized as unsatisfactory, but it should be suggested that the “fixed” step should probably be adjusted depending on the predicted folding behavior of the protein that is meant to be used as a sample (the user can adjust the code as they please). As stated above in this section, a stable peptide structure produces more

accurate results, but even considering these cases the results of the “Jude” method are overall “pessimistic”.

5. Conclusions

5.1 The classic “Good-Turing” method and the “max of mins” method

The classic “Good-Turing” method is the only existing method that analyses an MD simulation and can answer whether convergence has been achieved or not with a probability. The results are very promising, as it looks like smaller matrices can provide reliable results regarding the sample’s convergence. The “max of mins” method is a method that could work as an interesting alternative, but the results from the “max of mins” method appear to be more “optimistic” than the ones deriving from the classic “Good-Turing” method–which was not expected. Also, even if the “optimism” of the “max of mins” method was not an issue, the “max of mins” method is a method that could work as an adequate alternative, so long as, the method can be applied independently from the classic “Good-Turing” method, (meaning that the desirable program will have the “max of mins” logic but it will actually be cost-effective). Overall, it looks like the “max of mins” logic could be an accurate enough and computationally cheaper alternative to the classic “Good-Turing” method, but as it has been stated above, the consistently more “optimistic” results of the “max of mins” method are an interesting-and not really expected-outcome.

5.2 The “independent” method

With the “independent” method there is no need for the program to calculate the whole matrix, since as the name itself states the method can be applied independently from the classic “good-turing” method. Analysis of the data so that the optimal factor can be determined, never takes place, which is the key change in the method that reduces the computational cost significantly. The

results that occur from the “independent” method are promising, since they provide an idea regarding “What is the ideal step that should be used for the analysis of peptide structures?”, but they are not adequate, because up until this point there is no formula for finding the ideal step for the different proteins, other than “trial and error”. The “independent” method indicates the potential of a cheaper alternative method for determining convergence with a probability (a probability of unobserved conformations for different RMSD values).

5.3 The “Jude” method

Out of every experimental method tested in this research, the “Jude” method is the most complete out of all of them. The “max of mins” method already “knows” which is the optimal sampling factor, for each peptide structure used as sample, thanks to the run of the classic “Good-Turing” method that happens alongside with the “max of mins” method, which is why the “max of mins” results are so robust and also why the method is not cost effective. On the other hand, the potential accuracy of the “independent” method is completely random, with the “ideal” step (stride) between frames determined only through comparison with the classic “Good-Turing” method. The “ideal” steps (strides) between frames determined through “trial and error” for the cost-effective, but unreliable “independent” method were taken into consideration for the creation of the “Jude” method. The “Jude” method has a “fixed” step (stride) between frames set initially in the program, so there is no need for the maxRMSD vs sampling factor distribution, lowering that way the computational cost. This approach is also better than the “independent” method, not only because the step (stride) between frames is not random, but because there are extra parameters used in order to determine whether the simulation time is enough that all important configurations for the analysis in hand have been observed and this

set of data can be obtained by the user, so additional conclusions can be deducted from them, apart from the final $P_{unobserved}$ vs RMSD distribution produced. What makes the “Jude” method superior to the previously tested ones, is the consistently “pessimistic” results it produces, ensuring that the question of whether convergence has been achieved can be answered with a probability that will more likely be higher and not lower, than that emitted from the established classic “Good-Turing” method. Ultimately, the “Jude” method is an adequate method for users who want to evaluate the convergence rates of peptide structures without resorting to extremely uneconomical methods or unsatisfactory ones that do not provide a probability.

6. List of abbreviations

MD: Molecular Dynamics

ET_A: Endothelin A

ET_B: Endothelin B

TM: Transmembrane

GPCRs: G-protein-coupled receptors

ET-1: Endothelin-1

IAPP: Islet Amyloid Polypeptide

CDP-1: Cysteine Deleted Protegrin-1

PG-1: Protegrin-1

NAT: N-terminal acetyltransferase

RMSD: Root Mean Square Deviation

Rop: Repressor of primer

7. References

1. Scott A. Hollingsworth, Ron O. Dror. n.d. ‘Molecular Dynamics Simulation for All’. doi:10.1016/j.neuron.2018.08.011.
2. Panagiotis I. Koukos and Nicholas M. Glykos. n.d. ‘On the Application of Good-Turing Statistics to Quantify Convergence of Biomolecular Simulations’. 209−217. doi:dx.doi.org/10.1021/ci4005817.
3. Alan Grossfield and Daniel M. Zuckerman. 2009. *Quantifying Uncertainty and Sampling Quality in Biomolecular Simulations*. Vol. 5.
4. Darrin M. York, Lee G. Pedersen, and Tom A. Darden. 1993. ‘The Effect of Long‐range Electrostatic Interactions in Simulations of Macromolecular Crystals: A Comparison of the Ewald and Truncated List Methods’. 99:8345–48. doi:https://doi.org/10.1063/1.465608.
5. Brandon D. Wilson and H. Tom Soh. 2020. ‘Re-Evaluating the Conventional Wisdom about Binding Assays’. 45(8):639–49. doi:10.1016/j.tibs.2020.04.005.
6. S. Shahriari and L. Kadem. 2018. *Numerical Methods and Advanced Simulation in Biomechanics and Biological Processes*.
7. Xuewei Liu, Danfeng Shi, Shuangyan Zhou, Hongli Liu, Huanxiang Liu, and Xiaojun Yao. 2017. ‘Molecular Dynamics Simulations and Novel Drug Discovery’. 13(1). doi:https://doi.org/10.1080/17460441.2018.1403419.
8. Jacob D Durrant and J Andrew McCammon. 2011. ‘Molecular Dynamics Simulations and Drug Discovery’. doi:https://doi.org/10.1186/1741-7007-9-71.
9. Tomas Hansson, Chris Oostenbrink, and WilfredF van Gunsteren. 2001. ‘Molecular Dynamics Simulations’. 12(2):190–96. doi:https://doi.org/10.1016/S0959-440X(02)00308-1.
10. Katharina Vollmayr-Lee. 2020. ‘Introduction to Molecular Dynamics Simulations’. 88(5):401–22. doi:https://doi.org/10.1119/10.0000654.
11. William A. Gale and Geoffrey Sampson. 2008. ‘Good‐turing Frequency Estimation without Tears’. 2(3):217–37. doi:https://doi.org/10.1080/09296179508590051.
12. Jeremy C. Simpson. n.d. ‘Functional Assays’. 617–20.
13. Abdulnour Y. Toukmaji and John A. Board Jr. 1996. ‘Ewald Summation Techniques in Perspective: A Survey’. 95:73–92. doi:0010-4655/96/$15.00.

14. Jérôme Hénin, Tony Lelièvre, Michael R. Shirts, Omar Valsson, Lucie, and Delemotte. 2022. ‘Enhanced Sampling Methods for Molecular Dynamics Simulations’.
15. Faez Iqbal Khan, Dong-Qing Wei, Ke-Ren Gu, Md. Imtaiyaz Hassan, and Shams Tabrez. 2016. ‘Current Updates on Computer Aided Protein Modeling and Designing’. 85:48–62. doi:https://doi.org/10.1016/j.ijbiomac.2015.12.072.
16. Ozge Sensoy, Jose G. Almeida, Javeria Shabbir, Irina S. Moreira, and Giulia Morra. 2017. ‘Chapter 16 - Computational Studies of G Protein-Coupled Receptor Complexes: Structure and Dynamics’. 142:205–45. doi:https://doi.org/10.1016/bs.mcb.2017.07.011.
17. Peter W. Hildebrand, Alexander S. Rose, and Johanna K.S. Tiemann. 2019. ‘Bringing Molecular Dynamics Simulation Data into View’. 44(11):902–13. doi:10.1016/j.tibs.2019.06.004.
18. Christophe Bounaix Morand du Puch, Mathieu Vanderstraete, Stéphanie Giraud, Christophe Lautrette, Niki Christou, and Muriel Mathonnet. 2021. ‘Benefits of Functional Assays in Personalized Cancer Medicine: More than Just a Proof-of-Concept’. 11(19):9538–56. doi:10.7150/thno.55954.
19. Njål Foldnes and Steffen Grønneberg. 2017. ‘Approximating Test Statistics Using Eigenvalue Block Averaging’. 25(1):101–14. doi:https://doi.org/10.1080/10705511.2017.1373021.
20. Zhe Li, Wanli Kang, Hongbin Yang, Bobo Zhou, Haizhuang Jiang, Dexin Liu, Han Jia, and Jiaqi Wang. 2022. ‘Advances of Supramolecular Interaction Systems for Improved Oil Recovery (IOR)’. 301. doi:https://doi.org/10.1016/j.cis.2022.102617.
21. Thomas D. Pollard. 2010. ‘A Guide to Simple and Informative Binding Assays’ edited by Douglas Kellogg. 21(23):4057–4298. doi:https://doi.org/10.1091/mbc.e10-08-0683.
22. Shinya Honda, Toshihiko Akiba, Yusuke S. Kato, Yoshito Sawada, Masakazu Sekijima, Miyuki Ishimura, Ayako Ooishi, Hideki Watanabe, Takayuki Odahara, and Kazuaki Harata. 2008. ‘Crystal Structure of a Ten-Amino Acid Protein’. *Journal of the American Chemichal Society* 130(46). doi:https://doi.org/10.1021/ja8030533.
23. Onofrio Zirafi, Kyeong-Ae Kim, Ludger Ständker, Katharina B. Mohr, Daniel Sauter, Anke Heigele, Silvia F. Kluge, Eliza Wiercinska, Doreen Chudziak,

Rudolf Richter, Barbara Moepps, Peter Gierschik, Virag Vas, Hartmut Geiger, Markus Lamla, Tanja Weil, Timo Burster, Andreas Zgraja, Francois Daubeuf, Nelly Frossard, Muriel Hachet-Haas, Fabian Heunisc, Christoph Reichetzeder, Jean-Luc Galzi, Javier Pérez-Castells, Angeles Canales-Mayordomo, Jesus Jiménez-Barbero, Guillermo Giménez-Gallego, Marion Schneider, James Shorter, Amalio Telenti, Berthold Hocher, Wolf-Georg Forssmann, Halvard Bonig, Frank Kirchhoff, and Jan Münch. 2015. 'Discovery and Characterization of an Endogenous CXCR4 Antagonist'. *Cell Reports* 11(5):737–47. doi:https://doi.org/10.1016/j.celrep.2015.03.061.

24. Glykos, N., Cesareni, G., and Kokkinidis, M. 1999. 'ALANINE 31 PROLINE MUTANT OF ROP PROTEIN'. https://doi.org/10.2210/pdb1B6Q/pdb.
25. D Zarena, Biswajit Mishra, Tamara Lushnikova, Fangyu Wang, and Guangshun Wang. 2017. 'The π Configuration of the WWW Motif of a Short Trp-Rich Peptide Is Critical for Targeting Bacterial Membranes, Disrupting Preformed Biofilms and Killing Methicillin-Resistant Staphylococcus Aureus'. *Biochemistry* 53(31):4039–43. doi:10.1021/acs.biochem.7b00456.
26. Olympia-Dialekti Vouzina, Alexandros Tafanidis, and Nicholas M. Glykos. n.d. 'The Curious Case of A31P, a Topology-Switching Mutant of the Repressor of Primer Protein: A Molecular Dynamics Study of Its Folding and Misfolding'. doi:https://doi.org/10.1021/acs.jcim.4c00575.
27. Harini Mohanram and Surajit Bhattacharjya. 2014. 'Cysteine Deleted Protegrin-1 (CDP-1): Anti-Bacterial Activity, Outer-Membrane Disruption and Selectivity'. *Biochimica et Biophysica Acta (BBA) - General Subjects* 1840(10):3006–30016. doi:https://doi.org/10.1016/j.bbagen.2014.06.018.
28. Wataru Shihoya, Tomohiro Nishizawa, Akiko Okuta, Kazutoshi Tani, Naoshi Dohmae, Yoshinori Fujiyoshi, Osamu Nureki, and Tomoko Doi. 2016. 'Activation Mechanism of Endothelin ETB Receptor by Endothelin-1'. *Nature* 537:363–68. doi:https://doi.org/10.1038/nature19319.
29. Angela B Soriaga, Smriti Sangwan, Ramsay Macdonald, Michael R Sawaya, and David Eisenberg. 2016. 'Crystal Structures of IAPP Amyloidogenic Segments Reveal a Novel Packing Motif of Out-of-Register Beta Sheets'. 120(26):5810–16. doi:10.1021/acs.jpcb.5b09981.

30. Sunbin Deng and Ronen Marmorstein. 2020. 'Protein N-Terminal Acetylation: Structural Basis, Mechanism, Versatility, and Regulation'. *Trends in Biochemical Sciences* 46(1):15–27. doi:10.1016/j.tibs.2020.08.005.
31. Venkata S. Mandala, Matthew J. McKay, Alexander A. Shcherbakov, Aurelio J. Dregni, Antonios Kolocouris, and Mei Hong. 2020. 'Structure and Drug Binding of the SARS-CoV-2 Envelope Protein Transmembrane Domain in Lipid Bilayers'. *Nature Structural & Molecular Biology* 27:1202–8. doi:https://doi.org/10.1038/s41594-020-00536-8.
32. Antonios Kolocouris, Isaiah Arkin, and Nicholas M. Glykos. 2022. 'A Proof-of-Concept Study of the Secondary Structure of Influenza A, B M2 and MERS- and SARS-CoV E Transmembrane Peptides Using Folding Molecular Dynamics Simulations in a Membrane Mimetic Solvent'. *Physical Chemistry Chemical Physics* 24:25391–402. doi:DOI https://doi.org/10.1039/D2CP02881F.
33. Kato, Y., Ishimura, M., and Honda, S. 2015. 'NMR STRUCTURE of A MUTANT OF CHIGNOLIN, CLN025'. https://www.wwpdb.org/pdb?id=pdb_00002rvd.
34. Perez-Castells J., Canales A., Jimenez-Barbero J., and Gimenez-Gallego G. 2025. 'Three Dimensional Structure of EPI-X4, a Human Albumin-Derived Peptide That Regulates Innate Immunity through the CXCR4/CXCL12 Chemotactic Axis and Antagonizes HIV-1 Entry'. https://doi.org/10.2210/pdb2N0X/pdb.
35. Wang G. and Zarena D. 2020. 'NMR Structure of WW291'. https://doi.org/10.2210/pdb6NM2/pdb.
36. Mohanram H. and Bhattacharjya S. 2014. 'Cysteine Deleted Protegrin-1 (CDP-1): Anti-Bacterial Activity, Outer-Membrane Disruption and Selectivity'. https://doi.org/10.2210/pdb2MQ2/pdb.
37. Soriaga A.B., Macdonald R., Sawaya M.R., Sangwan S., and Eisenberg D. 2015. 'Structure of Amyloid Forming Peptide NFGAILS (Residues 22-28) from Islet Amyloid Polypeptide'. https://doi.org/10.2210/pdb5E5V/pdb.
38. Mandala, V.S., Hong, M., McKay, M.J., Shcherbakov, A.S., and Dregni, A.J. 2020. 'SARS-CoV-2 Envelope Protein Transmembrane Domain: Pentameric Structure Determined by Solid-State NMR'. https://doi.org/10.2210/pdb7K3G/pdb.

39. Thomaston, J.L. 2015. ‘Influenza A M2 Wild Type TM Domain at High pH in the Lipidic Cubic Phase under Cryo Diffraction Conditions’. https://doi.org/10.2210/pdb4QK7/pdb.
40. Shihoya, W., Nishizawa, T., Okuta, A., Tani, K., Fujiyoshi, Y., Dohmae, N., Nureki, O., and Doi, T. 2026. ‘Human Endothelin Receptor Type-B in Complex with ET-1’. https://doi.org/10.2210/pdb5GLH/pdb.
41. Glykos, N.M. and Kokkinidis, M. 2002. ‘ALANINE 31 PROLINE MUTANT OF ROP PROTEIN, MONOCLINIC FORM’. https://doi.org/10.2210/pdb1GMG/pdb.
42. Prior, S.H. 2019. ‘Trpzip2 Structure in Presence of Exogenous Haloprotectant Molecule.’ https://doi.org/10.2210/pdb6H7I/pdb.
43. Bhunia, A. and Datta, A. 2016. ‘Solution Structure of VG16KRKP in C.Neoformans (Conformation 1)’. https://doi.org/10.2210/pdb2N9N/pdb.
44. Koukos, P.I. & Glykos, N.M. (2013), "grcarma: A Fully Automated Task-Oriented Interface for the Analysis of Molecular Dynamics Trajectories", *J. Comput. Chem.*, **34**, 2310-2312
45. Glykos, N.M. (2006), "Carma: a molecular dynamics analysis program", *J. Comput. Chem.*, **27**, 1765-1768
46. Serafeim, A.-P., Salamanos, G., Patapati, K.K. & Glykos*, N.M. (2016), “Sensitivity of Folding Molecular Dynamics Simulations to Even Minor Force Field Changes”, J. Chem. Inf. Model., 56, 2035-2041. doi: 10.1021/acs.jcim.6b00493.