

A new method for protein characterization and classification using geometrical features for 3D face analysis: An example of tubulin structures

*Original*

A new method for protein characterization and classification using geometrical features for 3D face analysis: An example of tubulin structures / Di Grazia, L., Aminpour, M., Vezzetti, E., Rezania, V., Marcolin, F., Tuszynski, J.A.. - In: PROTEINS. - ISSN 0887-3585. - ELETTRONICO. - 89:1(2021), pp. 1-15. [10.1002/prot.25993]

*Availability:*

This version is available at: 11583/2847569 since: 2025-02-14T10:32:10Z

*Publisher:*

John Wiley and Sons Inc.

*Published*

DOI:10.1002/prot.25993

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

Wiley postprint/Author's Accepted Manuscript

This is the peer reviewed version of the above quoted article, which has been published in final form at <http://dx.doi.org/10.1002/prot.25993>. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

(Article begins on next page)

1 **A new method for protein characterization and**  
2 **classification using geometrical features for 3D face**  
3 **analysis: an example of tubulin structures**

4  
5  
6 Luca Di Grazia <sup>1</sup>, Maral Aminpour <sup>2,3</sup> Enrico Vezzetti <sup>1</sup>, Vahid Rezania <sup>4</sup>, Federica  
7 Marcolin <sup>1</sup>, and Jack Adam Tuszynski <sup>1,2,3\*</sup>

8  
9  
10 <sup>1</sup> Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino, Italy;

11 <sup>2</sup> Department of Physics, University of Alberta, Edmonton, Alberta, Canada

12 <sup>3</sup> Department of Oncology, University of Alberta, Edmonton, Canada

13 <sup>4</sup> Department of Physical Sciences, MacEwan University, Edmonton, Alberta, Canada

14 \* Correspondence: jacek.tuszynski@polito.it;

15  
16  
17  
18 **Short title/running title:** protein characterization and classification using geometrical  
19 features for 3D face analysis

45 **Abstract:** This paper reports on the results of research aimed to translate biometric 3D  
46 face recognition concepts and algorithms into the field of protein biophysics in order to  
47 precisely and rapidly classify morphological features of protein surfaces. Both human  
48 faces and protein surfaces are free-forms and some descriptors used in differential  
49 geometry can be used to describe them applying the principles of feature extraction  
50 developed for computer vision and pattern recognition. The first part of this study focused  
51 on building the protein dataset using a simulation tool and performing feature extraction  
52 using novel geometrical descriptors. The second part tested the method on two examples,  
53 first involved a classification of tubulin isotypes and the second compared tubulin with  
54 the FtsZ protein, which is its bacterial analogue. An additional test involved several  
55 unrelated proteins. Different classification methodologies have been used: a classic  
56 approach with a Support Vector Machine (SVM) classifier and an unsupervised learning  
57 with a k-means approach. The best result was obtained with SVM and the radial basis  
58 function (RBF) kernel. The results are significant and competitive with the state-of-the-  
59 art protein classification methods. This leads to a new methodological direction in protein  
60 structure analysis.

61 **Keywords:** 3D Face Analysis; Protein Classification; Tubulin; SVM; Geometrical  
62 Descriptors; Differential Geometry; Machine Learning

63

---

64 **1. Introduction**

65 The structure of a protein is an important indicator of its potential biological functions,  
66 especially its surface, which is exposed to the solvent and participates in interactions with  
67 other proteins and ligands. In a recently published work [1] it was shown how to capture

68 fingerprints of a protein using deep learning methodology and a strong correlation was  
69 demonstrated between the structure of a protein and its biological behavior. Another work  
70 [2] showed the relevant role of protein-protein interactions using local structural features.  
71 In this latter paper geometrical features were found to be interesting in this context.

72       The first step in the process of classifying proteins is to acquire a realistic (usually  
73 experimental) 3D dataset regarding a protein's structure. X-ray crystallography has made  
74 the largest and most important contribution to our understanding of protein  
75 structure. Nuclear Magnetic Resonance (NMR) and cryogenic electron microscopy (cryo-  
76 EM) are other methods by which to determine the protein structure [3] but they have  
77 various limitations. As an alternative to crystallographic structure determination, a  
78 computational method can be used to generate its prediction using a three-dimensional  
79 model [4]. However, proteins are non-static molecular structures, thus a crystallography-  
80 generated image is only a snapshot in time of a protein structure and not a fully realistic  
81 representation of all protein states, which can be quite dynamic. Therefore, molecular  
82 dynamics (MD) is a useful computational tool that can be used to produce atomic  
83 coordinate trajectories in order to provide a sampling of structural representations of a  
84 given protein. The method we propose in this paper is agnostic to the origin of the data,  
85 which in the case of proteins can either be obtained from experiments such as cryo-EM or  
86 synthetically generated from computational approaches such as MD. The key aspect is to  
87 have an atomistic model of the objects studied [3], which serves as the starting point for  
88 feature extraction based on the protein surface. Such a model provides a high-resolution  
89 representation of the object of interest, which is later on processed and characterized by a  
90 manageable number of parameters.

91 A protein can have different equilibrium conformational states that depend on ambient  
92 conditions. Moreover, some proteins are expressed by several genes leading to different  
93 isotypes with a high degree of structural similarity making accurate comparison important,  
94 so a dataset with significant number of different frames is important in order to have a  
95 statistically significant and valid test set. The most difficult task would be to distinguish  
96 between very closely related proteins or indeed the same protein in its wild type form and  
97 a mutated protein structure. For clearly distinct protein structures, standard approaches for  
98 their comparisons such as the use of the RMSD (root mean squared deviation) may work  
99 reasonably well but providing a single parameter only for structure comparisons may not  
100 always be useful or sensitive enough to distinguish subtle structural changes involving, for  
101 example, single point mutations or a small number of amino acid substitutions. It should  
102 also be mentioned that while sequence comparison methods are rapid and reliable, since  
103 there is no general solution to the protein folding problem, sequence comparisons are  
104 insufficient by themselves to inform us about subtle structural changes that can distinguish  
105 between highly similar protein structures.

106 Some experimentation has already been undertaken to classify proteins according to  
107 their states. Tsuda et al. adopted a Support Vector Machine (SVM) classifier for fast protein  
108 classification [5]. They obtained 13 classes and reached an accuracy of about 90%. Weston  
109 et al. [6] used a semi-supervised classification with a kernel cluster and reached a result of  
110 94.3%. Another interesting result has been obtained using a random forest approach and  
111 fifteen different supervised methods with about 11,000 pairs of protein domains leading to  
112 an accuracy of 97.0% [7]. Our focus in this paper is on accurate differentiation between  
113 structurally-similar proteins, which is a much harder problem to solve than comparing

114 vastly different protein structures. Many cases of protein families can be found and it is  
115 important to be able to find characteristic features distinguishing proteins belonging to the  
116 same family. This could be valuable with respect to their functional roles in cell biology as  
117 well as potential applications in rational drug design.

118 One of the most important proteins abundantly expressed in all eukaryotic cells is the  
119 family of tubulin proteins, which is studied in this paper as a challenging test case for this  
120 methodology. It is also highly homologous with its bacterial ancestor, FtsZ, which will also  
121 be used here for comparison. We should stress again that comparing protein sequences is  
122 a trivial problem in bioinformatics while 3D structural features of folded proteins pose a  
123 much greater challenge, which is addressed here.

124 In the computational experiment reported below SVM was used because the quantity  
125 of data tested was relatively low, and a deep learning approach requires large data sets to  
126 achieve a high level of confidence. The novelty of our approach rests with the feature  
127 extraction using geometrical descriptors and its general applicability to 3D structure  
128 characterization, because geometric feature surfaces were used with significant results in  
129 many other applications before, e.g. [8, 9]. We believe that the classification provided here  
130 can be further improved with more data, more classes and a complex neural network. A  
131 complex neural network is one of the applications we are planning to implement in the near  
132 future. We intend to use a convolutional neural network to minimize the cost function to  
133 cluster the inputs correctly, because this could be an efficient way to find a pattern in the  
134 input data and it can be a significant improvement for our objectives. All of which is  
135 planned for future work, especially within the context of geometric deep learning [10],  
136 which nowadays is the state-of-the art of classification.

137 Tubulin is a key cytoskeletal protein, which has been exhaustively studied for its  
138 applications in several fields, including (i) being the target for various anti-cancer drugs  
139 [11] and (ii) the discrimination of the Saccharomyces complex [12]. It is a globular protein  
140 with a molecular weight of 55 kDa per monomer and its numerous isotypes expressed by  
141 separate genes have a broad distribution in animal and plant cells [13]. Tubulin is a building  
142 block of microtubules (MTs) and its stable form is an  $\alpha\beta$  -heterodimer. MTs play various  
143 important roles in all eukaryotic cells including cell motility, material transport and most  
144 importantly cell division where MTs form mitotic spindles [12-13].

145 The novelty of the present work rests with the application of geometrical descriptors  
146 coming from the field of face analysis to the classification of surfaces of proteins, with the  
147 aim of adopting this geometrical information as descriptive features and discriminating  
148 elements to classify proteins. Here, we test the method on the examples of tubulin isotypes  
149 and related proteins (e.g. FtsZ). The method can, of course, be applied to an arbitrary  
150 protein or indeed a protein complex but being able to discriminate between highly  
151 homologous proteins based on the geometrical shapes of their surfaces opens the door to  
152 numerous applications across the field of protein science. The idea comes from the  
153 realization that geometrical properties can well describe the surface of a 3D object such as  
154 a protein and could identify characteristic features when comparing two or more similar  
155 structures. Proteins surfaces can be split into two outer surfaces by cutting a plane through  
156 the data set including the main axis of rotational symmetry. These two halves of the outer  
157 surface, similarly to human faces, differ from one another depending on the protein type,  
158 and also can change their conformational states dynamically, similarly to human facial  
159 expressions. Thus, what in the field of pattern recognition is called face recognition could

160 be transferred to the context of protein classification according to the typology. These  
161 common points have fostered the interest of uncovering the potentiality of cross-  
162 fertilization between these two fields with the aim of better categorization.

163 All eukaryotic organisms carry multiple genes coding for  $\alpha$  and  $\beta$  tubulin (and other  
164 variants, e.g.  $\gamma$ ), which are referred to as isoforms when comparing tubulin expressed by  
165 different organisms. When a single organism is discussed, various tubulin genes code for  
166 what are called tubulin isotypes. Isotypes have highly homologous amino acid sequences  
167 that appear to have diverged as a result of accumulated mutations since their separation by  
168 distinct speciation events [14]. Amino acid sequence similarity is very high for all tubulin  
169 proteins both within and between diverse species making structural comparisons difficult.  
170 At the cellular level, the roles of the  $\alpha$  and  $\beta$  tubulin isotypes are essential, a result of subtle  
171 structural variations within their sequences [15] Several isotypes of the  $\alpha$  and  $\beta$  tubulins  
172 have been identified in human cells, their existence and distribution providing a link to  
173 their specific roles in the polymerization and stability of MTs, among other roles [8]  
174 making structural differences correlate with functional roles in cells, importantly including  
175 cancer cells. For example,  $\beta$ II tubulin has been a common target for chemotherapy drug  
176 action and is involved in protein-protein interactions [2]. Hence again, the structural  
177 differences between tubulin isotypes significantly assist in drug design targeting specific  
178 isotypes such as  $\beta$ III, which is overexpressed in all cancer cells. Through a search of  
179 available protein sequence databases, a total of ten unique  $\beta$  tubulin isotypes can be found,  
180 all of which have highly similar amino acid sequences and are generally well conserved.  
181 Sequence alignment, similarity and identity values of the studied isotype proteins (see  
182 below for details) range between 78% and 98%, indicating a major level of similarity

183 between these structures. The question that remains is how do these sequence variations  
184 translate into structural differences.

185 As stated above, MTs are dynamic cytoskeleton polymers present in all eukaryotic  
186 cells made up of the protein tubulin. FtsZ is a close structural homologue of tubulin within  
187 prokaryotic cells, and plays an important functional role during bacterial cell division. A  
188 close relationship between FtsZ and tubulin can be seen from their very similar protein  
189 structures (Figure 1a). Both  $\alpha$  and  $\beta$  tubulin share an approximate 35% sequence identity  
190 with FtsZ [16]. Both FtsZ and tubulin can assemble to form straight filaments. This  
191 association is regulated by guanosine triphosphate (GTP), which is bound in the junction  
192 between adjacent monomers (Figure 1b). FtsZ forms long protofilaments consisting of a  
193 single string of FtsZ proteins in contrast to tubulin, which makes cylindrical MTs. Unlike  
194 tubulin, FtsZ does not appear to provide a structural role throughout the bacterial cell cycle,  
195 but instead just plays a structural role during bacterial cell division, when it forms a band,  
196 known as the Z-ring, around the inner cell wall at the location where the cell will divide.

### 197 **Figure 1**

198 The main goal of the research reported here has been to investigate the following issues:

- 199 • whether it is possible to rely on features coming from the field of pattern  
200 recognition and face analysis to geometrically describe (and classify) the  
201 geometrical properties of the protein surface;
- 202 • whether it is possible to recognize different isotypes of the same protein from a  
203 different set of molecular dynamics snapshots;

- 204       • whether it is possible distinguish between two highly structurally similar but not  
205       identical proteins such as tubulin and FtsZ, and whether it is possible to distinguish  
206       arbitrary proteins with no relation to each other.

207 It is worth stating in this context that in general the main goal of a classifier is to separate  
208 objects belonging to different classes using a number of possible linear separators as shown  
209 in the examples presented in Figure 2.

210 **Figure 2**

211       It is reasonable to expect that using one of these separators one can get a datum that  
212 is on the other side of the hyperplane, which would then be misclassified because the  
213 hyperplane is really near the ham data [17]. SVM is able to find a solution with a larger  
214 margin for the two-separator classifier as shown in Figure 2(a). This hyperplane works  
215 better than others as it is expected to reduce the number of misclassifications, because it is  
216 the one with the highest margins from the two sets of data.

217 The first part of this paper describes the development of the dataset using tubulin isotypes  
218 and FtsZ protein as test cases. Then, geometrical descriptors are computed on the 3D  
219 surface of these proteins. They are then converted into histograms and saved in a file. This  
220 file is the input of the classifiers. The code is provided in a pCloud repository [18]. The  
221 entire process is summarized in Figure 3.

222 **Figure 3**

223       This paper is organized as follows. In Section 2 geometrical descriptors used for  
224 implementing the feature extraction are described. Section 3 is the core of the paper and it  
225 outlines feature extraction and classification methods with a detailed description of the  
226 strategies and techniques performed. Section 4 summarizes and discusses the results

227 comparing them with the-state-of-art results. Finally, Section 5 summarizes the work and  
228 discusses future developments.

## 229 **2. Geometrical descriptors**

230 The surfaces representing both human faces and proteins are geometrically considered  
231 as a free form. Thus, features coming from the field of differential geometry can be applied  
232 in order to understand their local and global properties. Geometrical descriptors are widely  
233 used in the area of 3D face recognition with significant results reported elsewhere in the  
234 literature [19, 20]. They underline different characteristics of a free-form and are an  
235 important tool for feature extraction [21] within the context of face analysis [22]. In this  
236 work, for the first time we apply these descriptors to proteins and use them for structural  
237 classification purposes [19].

238 The geometrical descriptors used in this research are the following geometrical  
239 descriptors [22, 23]: mean curvature ( $H_{mean}$ ), principal curvatures ( $k_{1_{mean}}$  and  $k_{2_{median}}$ ),  
240 the shape index ( $S_{mean}$ ), the third coefficient of the second fundamental form ( $g_{mean}$  and  
241  $sing$ ), and a descriptor enlightening the symmetry property ( $F_{den2}$ ). Considering that these  
242 descriptors rely on the derivatives of the surface ( $h_x, h_y$ ), they well describe the changes in  
243 surface curvature ( $k_{1_{mean}}, sing, k_{2_{median}}, g_{mean}, H_{mean}$ ), depressions and peaks (local  
244 minima and maxima) of the surface ( $k_{1_{mean}}, sing, k_{2_{median}}, g_{mean}, H_{mean}$ ), the shapes  
245 in terms of the types of surfaces ( $S_{mean}$ ), and the surface's symmetry property ( $F_{den2}$ ).  
246 These parameters are highly informative of the investigated surface's geometrical  
247 properties. Each descriptor can underline a specific characteristic of a certain surface.  
248 These descriptors are briefly described below in regard to their conceptual order. The first

249 and second fundamental forms provide the first six descriptors of the set. They are used to  
250 measure distance on surfaces and are defined by the formula

$$251 \quad ds^2 = Edu^2 + 2Fdudv + Gdv^2 \quad (1)$$

252 where  $E, F, G, e, f$  and  $g$  are their coefficients given by:

253

$$254 \quad E = 1 + h_x^2, \quad (2)$$

$$255 \quad F = h_x h_y, \quad (3)$$

$$256 \quad G = 1 + h_y^2, \quad (4)$$

$$257 \quad e = \frac{h_{xx}}{\sqrt{1+h_x^2+h_y^2}}, \quad (5)$$

$$258 \quad f = \frac{h_{xy}}{\sqrt{1+h_x^2+h_y^2}}, \quad (6)$$

$$259 \quad g = \frac{h_{yy}}{\sqrt{1+h_x^2+h_y^2}}. \quad (7)$$

260 Curvatures are used to measure how a regular surface  $x$  bends in. If  $D$  is the differential  
261 and  $N$  is the normal plane to a surface, then the determinant of  $DN$  is the product of the  
262 principal curvatures, and the trace of  $DN$  is the negative of the sum of principal curvatures.  
263 At point  $P$ , the determinant is the Gaussian curvature  $K$  of  $x$  at  $P$ . The negative of half of  
264 the trace of  $DN$  is called the mean curvature  $H$  of  $x$  at  $P$ .

265 The principal curvatures  $k_1, k_2$  are the roots of the quadratic equation given below:

$$266 \quad x^2 - 2Hx + K = 0 \quad (8)$$

267 Thus, we can choose  $k_1$  and  $k_2$  so that:

$$268 \quad k_1 = H + \sqrt{H^2 - K} \text{ and } k_2 = H - \sqrt{H^2 - K} \quad (9)$$

269 where

270 
$$K = \frac{eg-f^2}{EG-F^2} \text{ (10)}$$

271 
$$H = \frac{eG - 2fF + gE}{2(EG - F^2)} \text{ (11)}$$

272 In terms of the principal curvatures, Gaussian ( $K$ ) and mean curvatures ( $H$ ) can be written  
 273 as

274 
$$K = k_1 k_2, \text{ (12)}$$

275 
$$H = \frac{k_1 + k_2}{2} \text{ (13)}$$

276 where  $h$  is a differentiable function representing the three-dimensional surface.

277 The shape index  $S$ , which describes the shape of the surface, is defined as [24]:

278

279 
$$S = -\frac{2}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2}, S \in [-1, 1], k_1 \leq k_2. \text{ (14)}$$

280

281 Some descriptors highlight particular facial lines, such as  $F_{den2}$ , which shows visible facial  
 282 part contours. It can be computed using the formula:

283 
$$\frac{F}{1 + (h_x)^2 + (h_y)^2}, \text{ (15)}$$

284 where:

- 285 •  $h$  is the differentiable function  $z = h(x, y)$  representing the face/protein surface;
- 286 •  $h_x$  and  $h_y$  are the first derivatives of  $h$  with respect to  $x$  and  $y$  [25].

287 In a protein,  $F_{den2}$  can underline different trends of the free form analyzed. In particular,  
 288 this descriptor has high and low values in correspondence to concavities and convexities,  
 289 and values approximately equal to zero on critical points.

290 The surfaces of human faces are given by depth maps, which are manageable as  
291 matrixes (X Y Z). For each coordinate pair X, Y, there is a unique value of Z. Since proteins  
292 do not have a default form, their surfaces are split up in two parts divided into two opposite  
293 faces: surfaces with a positive Z-axis and those with a negative Z-axis in order to yield two  
294 shells that complete the protein surface.

295 The descriptors used are mapped onto the surfaces as described in Section 3.4. These  
296 descriptors are calculated for all protein faces considered in the following. An  
297 example of  $F_{den2}$  applied to both a human face and a protein is shown in Figure 4a.  
298 The descriptor *sing* is built from the application of the sine standard function applied  
299 to the third coefficient of the second fundamental form ( $g$ ) (see Figure 4b) [23]. Mean  
300 and median filters have been applied to the primary descriptors  $S$ ,  $k_1$ ,  $k_2$ ,  $g$ , and  $H$ .  
301 Mean and median values are computed in squared neighborhoods of side 5 around  
302 each point of the facial depth maps [23]. These descriptors are labelled as follows:  
303  $S_{mean}$ , (see Figure 4c),  $k_{1mean}$  (see Figure 4d),  $k_{2median}$  (see Figure 4e),  $g_{mean}$  (see  
304 Figure 4f) and  $H_{mean}$  (see Figure 4g).

305 **Figure 4**

### 306 **3. Material and methods**

307 At the beginning of this section we give a brief introduction to some basic concepts  
308 related to Machine Learning, which can be useful for understanding the methods used in  
309 this paper. Machine Learning (ML) is a subset of Artificial Intelligence (AI) tools that  
310 include mathematical and statistical models, which complete tasks with experience gained  
311 through training. The quality and amount of the training data have an important role in this  
312 process. ML classifiers can be divided into two types based on their training methods:

313 supervised and unsupervised learning. Supervised learning needs a training phase with  
314 labeled training data (i.e. sample data containing input-output pairs) in order to learn the  
315 relationship between the input and output data. On the other hand, unsupervised learning  
316 algorithms do not employ labeled training data and they aim to divide the dataset into  
317 clusters without the training phase. In this work, we use a discriminative model (a  
318 supervised model) that employs Support Vector Machine (SVM). The aim of this model is  
319 to determine the division of different clusters without considering how data are generated,  
320 unlike generative models, which do consider how the data are generated during the process.  
321 In our model, dot-product kernels are used to compute the similarity between two vectors  
322 in a higher dimensional feature in a more efficient manner. For the SVM, we tried both  
323 linear and non-linear kernels. As the linear kernel essentially performs the normal dot-  
324 product, the similarity score is calculated as the length of the projection of one vector onto  
325 another. The non-linear kernel can perform the dot product in a higher dimensional feature  
326 space. Even though non-linear kernels may be slower to use due to the computational  
327 complexity, they usually yield more favorable results. Geometric Deep Learning is a new  
328 field in deep learning that aims to build neural networks that can learn from non-Euclidean  
329 data, for example from graphs or complex surfaces.

330 The process we follow in this paper starts with the collection of protein data. In the present  
331 example we focus on tubulin whose bovine structure has been crystallized and can be found  
332 in the Protein Data Bank (PDB). However, its various isotypes have not been crystallized  
333 and hence these structures need to be generated by homology modeling using the bovine  
334 (not human) variant of this protein as a template. To obtain frames of the protein structure,  
335 it is necessary to run MD simulations for some time, typically 10-100 nanoseconds and

336 take snapshots, approximately every nanosecond, at the very moment when the structure  
337 relaxes to an equilibrium conformation. Only the atoms comprising the protein are kept in  
338 the file used for these MD simulations with the ligand atoms removed in order to avoid  
339 false representations of the protein since ligands are not part of the protein and can form  
340 an occlusion during the process of protein recognition. The next step in this computational  
341 experiment is to analyze similar but not identical proteins and their states, for example  
342 tubulin isotypes with each other or a tubulin isotype and FtsZ and to compare the two for  
343 similarities and differences.

344 The result of these MD simulations is in each case a PDB-formatted file that is a 3D  
345 representation of a protein, which is converted into a MAT file using a MATLAB script.  
346 In the current work several software packages are used: Matlab 9.5 (R2018b) [26] for the  
347 feature extraction using geometrical descriptors, Anaconda 1.9.6 [27] with Python 3.7 [28]  
348 and the library sklearn 0.22 [29] for the implementation of classification methods and R-  
349 3.5.3 for the k-means algorithm [30].

### 350 **3.1. Molecular dynamics simulations**

351 The tubulin crystal structures available in the PDB are those for bovine protein. The  
352 bovine tubulin structure of tubulin (PDB ID: 1JFF) [31] was used as a template to construct  
353 the homology model for human  $\alpha\beta$  tubulin isotypes ( $\beta$ I (UniProtKb: P07437),  $\beta$ IIa  
354 (UniProtKb: UniProtKb: Q13885),  $\beta$ IIb (UniProtKb: Q9BVA1),  $\beta$ III (UniProtKb:  
355 Q13509),  $\beta$ IVa (UniProtKb: P04350),  $\alpha\beta$ IVb (UniProtKb: P68371),  $\alpha\beta$ V (UniProtKb:  
356 Q9BUF5),  $\alpha\beta$ VI (UniProtKb: Q9H4B7) and  $\beta$ VIII(UniProtKb: Q3ZCM7)) using the  
357 Molecular Operating Environment (MOE) software package [32]. Multiple sequence

358 alignment results contained in Figure 5 show that human  $\beta$ -tubulin isotypes exhibit residue  
359 composition variations at different locations.

360 **Figure 5**

361 Sequence similarity matrix and sequence identity matrix of the tubulin isotypes are  
362 shown in Figure 6(a) and (b), respectively. The matrix values (i, j) for the percentage  
363 identity and similarity metrics are equal to the number of sequence matches between chains  
364 i and j, divided by the number of residues in chain i. Residues are considered identical if  
365 their single-letter code is the same (note that MSE-Selenomethionine and MET-Methionine  
366 are considered "identical"). Residues are "similar" if their BLOSUM62 substitution score  
367 is greater than zero.

368 **Figure 6**

369 The atomic coordinates of similar but not identical FtsZ dimer were obtained from  
370 the Protein Data Bank as (PDB ID: 1W5B) [33]. The coordinates for the missing residues  
371 of the proteins were obtained by modeling using the MOE package [32]. Since the C-  
372 terminus has not been included in the electron crystallography data for the tubulin structure,  
373 we did not consider it in our calculations. The missing hydrogens for heavy atoms were  
374 added using the tLEAP module of AMBER [34] with the AMBER14SB force field. The  
375 protonation states of all ionizable residues were determined at pH = 7 using the MOE  
376 program. Each protein model was solvated in a 12 Å box of TIP3P water. Na<sup>+</sup> and Cl<sup>-</sup>  
377 ions were added in order to bring the salt concentration to the physiological value of 0.15  
378 M. After minimization, the MD simulations were carried out in three steps: heating, density  
379 equilibration, and production. First, each solvated system was heated to 300 K for 50 ps,  
380 with weak restraints on all backbone atoms. Next, density equilibration was carried out for

381 50 ps of constant pressure equilibration at 300 K, with weak restraints. Finally, MD  
382 production runs were performed on all systems for 100 ns. Ligands and ions were all  
383 removed from the complex after equilibration in order to avoid false representations of the  
384 protein since ligands can form an occlusion during the process of protein recognition. After  
385 equilibration, density-based clustering algorithm from the AMBER software was used for  
386 cluster analysis of MD trajectories (20). Several snapshots from top clusters were selected  
387 for all further calculations in the study.

388 The result of our simulation is a PDB-formatted file (a 3D representation of all atoms  
389 comprising the protein), which is converted into a MAT file using a MATLAB script.

### 390 **3.2. Data augmentation**

391 To expand the dataset for FtsZ, a data augmentation technique is used where each  
392 structure is rotated around the Z-axis in 40° steps. Subsequently, the 3D protein  
393 representation is ready to be used for feature extraction. It was not necessary to follow the  
394 same procedure for tubulin since we have many examples available. The purpose of  
395 reorienting the z-axis is not only to obtain additional examples, but also in order to not have  
396 a bias inside the classifier, in fact most of the rotated proteins were used during the test  
397 phase. Both hemispheres of the protein were used to have a complete dataset. Then, to  
398 avoid the over-fitting problem a k-fold cross validation is implemented with  $k = 5$ .

399 Cross validation is a powerful technique used to avoid overfitting. When the model is  
400 trained and tested on the same dataset, high scores can be easily obtained since the model  
401 becomes biased. In this case, low score results are obtained when the model is tested on  
402 an unseen dataset. Using cross validation, the dataset is divided into  $k$  sub parts, called  
403 folds. Then, the training is performed iteratively on the  $k-1$  folds and the remaining fold is

404 used for the testing phase. In this way, the test set will be a truly unseen dataset for the  
405 model. One such example is shown in Figure 7 (<https://pubs.nlm.nih.gov/>) [35].

406 **Figure 7**

407 At this point the 3D protein representation is ready and the feature extraction can be  
408 performed.

409 **3.3. Protein samples**

410 In this computational experiment, we used a total of 889 examples of tubulin structure files  
411 for 9 isotypes, as shown in Table 1.

412 **Table 1**

413 Using data augmentation, the 13 FtsZ protein samples were rotated in order to create 65  
414 samples, most of them used only during the test phase. The binary classification between  
415 tubulin and FtsZ was performed using the samples shown in Table 2.

416 **Table 2**

417 **3.4. Data processing**

418 The x-, y- and z-coordinates were extracted from the PDB file. First, the data were shifted  
419 in order to be geometrically symmetric with respect to x-, y- and z- axes, i.e. the center of  
420 the coordinate systems is the geometric center of the dataset:  $(x, y, z) \rightarrow (x - \Delta x, y - \Delta y, z$   
421  $- \Delta z)$  where  $\Delta x = (x_{\max} - x_{\min})/2$ ,  $\Delta y = (y_{\max} - y_{\min})/2$  and  $\Delta z = (z_{\max} - z_{\min})/2$ .  
422 Then, the data were divided into two groups of positive and negative z-values. Finally, for  
423 each group, the exterior surface with a desired resolution was calculated using "meshgrid"  
424 and "griddata" commands in Matlab with the cubic interpolation method.

425 The descriptors were mapped onto the surfaces as follows. The surfaces were given  
426 by point clouds where points are non-connected (not a mesh) and arranged in a square grid.

427 This type of data is called depth map and can be described by matrices: X, Y, Z, where Z  
428 is the one describing the “surface” and is represented in these formulas as  $h$ . Through  
429 Matlab “gradient” function, the derivatives  $h_x$ ,  $h_y$ ... were evaluated so that other matrices  
430 representing the first derivative with respect to  $x$ , the first derivative with respect to  $y$ , etc.,  
431 were generated and stored. Then, the implementation formulas for the descriptors were  
432 calculated on the matrices previously computed and new matrices were obtained  
433 representing every geometrical descriptor.

434 For each protein the Z axis was divided in two files: one for the positive part and  
435 the second for the negative part using the formula:  $z - \max(z) + (|\max(z) - \min(z)|)/2$ . Each  
436 part represents a “face” of the protein and the geometrical features were computed for both  
437 the faces. Then, for every geometrical descriptor a 9-bin histogram was created with the  
438 same equidistance for the X-axis.

439 The MATLAB code loaded all data and the following processing steps were performed for  
440 all the datasets:

- 441 • the class of the protein was extracted from the filename and the class was recorded  
442 in the first column of the dataset matrix;
- 443 • geometrical descriptors were computed from matrix Z (positive and negative);
- 444 • histograms were created and each bin was written in the right column of the dataset  
445 matrix;
- 446 • at the end of each loop the dataset matrix became the input for the classifier.

447 The entire process is summarized in Figure 8.

448 **Figure 8**

449 In this computational experiment, 9 isotypes were used (indeed, the classifier will work  
450 with 9 classes). The classes were chosen 1 to 9 in an ascendant order as shown in Table 3.

451 **Table 3**

452 This task was performed using a switch case construct. The right class was written in the  
453 first column of the Features Matrix.

454 **3.5. Feature extraction**

455 For every geometrical descriptor, a 9-bin histogram was created. Since it is possible  
456 that some descriptors have values  $\in \mathbb{C}$  (complex), a check was performed first. The  
457 geometrical descriptors were calculated using 9 bins and the X-axis values were  
458 compressed between -0.2 and 0.2, then the Y-axis values were saved and used as features.  
459 Some examples of histograms are shown in Figure 9.

460 **Figure 9**

461 Finally, when all descriptors for all protein data were computed, the resultant matrix  
462 was copied into a file. For tubulin and other proteins these descriptors can underline  
463 specific characteristic of a certain surface. They can indicate different trends of the free  
464 form analyzed and they can describe the shape of the surface. The features are extracted  
465 with multiple geometrical descriptors to extract more details; using this approach, also  
466 small differences in convexity and concavity can be recognized during the classification.  
467 Analyzing the features extracted, the most important features were found from parameter  
468 values of *Fden2* and *sing*, because analyzing the data these values were sufficiently  
469 different to help the classifier select the right class. In particular, *Fden2* is meant to be  
470 descriptive for the its behavior in the loci of critical points, and *sing* for curvature changes,  
471 local minimums in convexities and local maximums in concavities, respectively.

472 **3.6. Classification**

473           The adopted classifiers were k-means and SVM. First, an unsupervised method was  
474 tested (k-means) using 9 clusters and a limited number of iterations, then a supervised  
475 method (SVM) using linear and non-linear kernels was used. In these cases, it is not a  
476 simply binary classification, but there are many classes (9) and many features (more than  
477 100), so some distributions cannot separate the dataset in a linear way or with a linear  
478 separator as a high misclassification rate is reached. An interesting improvement is to use  
479 a non-linear separator or a kernel trick. An example of a non-linear kernel is the RBF  
480 kernel, which in this test led to positive results.

481 A linear and a nonlinear kernel (RBF in our case) were chosen in order to see whether a  
482 non-linear kernel can reach better results. The difference between linear and non-linear  
483 kernel is on the way they divided dataset into classes. A linear kernel uses a linear function  
484 to divide it and it is less time consuming but also less precise. A non-linear kernel uses a  
485 non-linear function, so it can divide the dataset better. The cross validation has not been  
486 performed here because the results were positive, and hence we have already avoided the  
487 overfitting problem. The validation part was performed using a large number of parameters  
488 and the best ones were selected for the testing part.

489 **3.6.1. k-means**

490           An unsupervised approach was performed using a k-means classifier implemented  
491 in R. The matrix file was loaded and the column with the label was deleted. Then, the  
492 classifier was tasked with finding 9 clusters in the input data and at the end there was a  
493 comparison made between the clustering and the right label.

494 k-means works in an iterative way and it performs three steps. In the first step, the dataset  
495 is loaded, and the number of clusters is chosen. The centroids are created in a random  
496 position. In the second step, each data point is assigned to a nearest cluster. The range for  
497 the initialization of the centroids of k-means is set from 2 to 10. The Euclidean distance is  
498 computed between a point and every centroid. The minimum distance centroid is chosen  
499 as the following cluster:

500 
$$\operatorname{argmin} \operatorname{dist}(c_i, x)^2,$$

501 where  $c$  is the centroid and  $x$  the data points. In this last phase the centroids are computed  
502 again as the mean of all the data points of the cluster:

503 
$$c_i = \frac{1}{|S_i|} \sum x_i,$$

504 where  $S_i$  is the sum of a single cluster. Therefore, new centroid positions are computed,  
505 and this loop continues until the centroid positions do not change significantly.

506 The stop condition is given by the following criteria:

- 507
- no data points change the cluster;
  - the sum of distances is at the minimum;
  - the maximum number of iterations is reached.
- 509

510 Therefore, when the convergence is obtained the algorithms stops.

511 The final result achieved in this example was 76.6%, which is an acceptable result,  
512 considering that it is an unsupervised method. Nonetheless, in order to improve the  
513 method's accuracy, other types of classifications were tested by us and we discuss them  
514 below.

### 515 **3.6.2. Support Vector Machine**

516 The first test was performed using a linear kernel where  $\lambda$  is a key parameter of SVM.  
517 In fact, the main factors in SVM are setting a large margin and reducing the  
518 misclassification rate. These two properties are inversely proportional, and the  $\lambda$  parameter  
519 helps to find a trade-off. A large value of  $\lambda$  is for a small margin, whereas a small value of  
520  $\lambda$  is for a large margin. The right  $\lambda$  parameter depends on the test data. The steps used are  
521 as follows:

- 522 • the dataset is loaded and features and labels are divided;
- 523 • the dataset is randomly split into 60% training set, 10 % validation set and 30% test  
524 set;
- 525 • the training is performed using a linear kernel. We then use different values of  $\lambda$  in  
526 the range  $10^{-5}$  to  $10^5$  and it is evaluated on the validation set. The best parameter  
527 found on the validation set is  $\lambda = 10^{-5}$  with a score of 95.1%;
- 528 the model is tested and scored on the validation set with the best parameters.

529 The accuracy obtained changes using different  $\lambda$  values. As a matter of fact, by  
530 increasing the  $\lambda$  value, the optimization will choose a smaller margin hyperplane, but the  
531 best parameters depend on the dataset and in this case the best value is obtained as  $\lambda =$   
532  $10^{-5}$ . The final evaluation on the test set with the best parameter  $\lambda = 10^{-5}$  was found to  
533 be 92.4%.

534 The dataset was built using 9 different Tubulin isotypes. Hence, the number of  
535 classes used for the SVM classifier was 9; the same number was used in the k-means test,  
536 in order to have comparable results. The confusion matrix is an important tool to evaluate  
537 the results, since it gives precise information about misclassification. A confusion matrix

538 without normalization and a normalized confusion matrix are represented in Figure 10. It  
539 this case, the accuracy is very high, since there is misclassification found only in one class.

540 **Figure 10**

541 The second test was performed using an RBF kernel. The number of features used was 112  
542 and the dataset was not large, so an approximation of the RBF kernel was not taken into  
543 consideration (22). The steps used are as follows:

- 544 • the dataset is loaded and features and labels are divided;
- 545 • the dataset is randomly split into 60% training set, 10 % validation set and 30% test  
546 set;
- 547 • the training is done using an RBF kernel. We then use different  $\lambda$  and gamma  
548 parameters in the range between  $10^{-5}$  to  $10^{15}$  and it is evaluated on the validation  
549 set. The best parameters on the validation set are found to be:  $\lambda = 100$  and gamma  
550  $= 10^{-9}$  with a score of 98.0%;
- 551 • the model is tested and scored on the validation set with the best parameters.

552 Note that the achieved accuracy changes significantly using different  $\lambda$  and gamma  
553 values. The gamma parameter that is used in the RBF kernel function is the inverse of the  
554 standard deviation of the RBF kernel, which is used as a similarity function. A small value  
555 of gamma indicates a large variance where two points can be matched as similar. This  
556 results in a smoother decision-making by the model. A higher gamma value has the  
557 opposite effect on the process. The challenge will be to find an optimum value of gamma  
558 for the given data set. Indeed, by increasing the  $\lambda$  value, the optimization will choose a  
559 smaller margin hyperplane, but the best parameter depends on the dataset selected and, in

560 this case, the best is 100. The final evaluation on the test set with the best parameter  $\lambda =$   
561 100,  $\gamma = 10^{-9}$  and the accuracy obtained was 96.5%.

562 The same methodology was applied to tubulin and FtsZ classifications.

#### 563 **4. Results and discussion**

564 In the case of tubulin isotype comparison, the best result was given by the SVM classifier  
565 with an RBF kernel. All results are summarized in Table 4.

#### 566 **Table 4**

567 In the case of tubulin and FtsZ comparison, the best result is also given by the SVM  
568 classifier with an RBF kernel. All results are summarized in Table 5.

#### 569 **Table 5**

570 These results are competitive with the state-of-the-art results found in the literature.  
571 A fast protein classification method [5] based on an SVM classifier reached an accuracy of  
572 about 90% with 13 classes. Another study [7] used a semi-supervised classification with a  
573 kernel cluster and achieved a 94.3 % accuracy. Consequently, the results of the present  
574 study appear to be significant. This work is a starting point toward protein classification  
575 based on geometrical features and we expect that even better results can be reached in the  
576 future. A natural continuation of this work can be to study important features of a protein,  
577 for example characterization of a binding pocket [36] for a ligand, a catalytic domain  
578 recognition or a protein-protein interaction interface.

579 A larger experiment was performed using several additional proteins in order to  
580 provide an increased validation for the method proposed in this paper. This test involved  
581 four arbitrarily chosen FtsZ protein structures, namely: 2R6R, 2VAW, 2VAP and 2VAM.  
582 These structures correspond, respectively, to the following biological species: *B. subtilis*,

583 *Pseudomonas aeruginosa*, *M. jannaschii* and *Aquifex aeolicus*. In this test 683 samples  
584 were used as listed in Table 6.

585 **Table 6**

586 The results of this test are very encouraging as shown in Table 7, which summarizes the  
587 use of various classifiers for different tests performed and their accuracy levels achieved.

588 **Table 7**

589 To avoid over-fitting and to generalize the method in a better way a 5-fold cross validation  
590 is performed. In this way the classifier is not biased by the test set and it also works well  
591 with other proteins. The last experiment showed that it also works well with four very  
592 different proteins. In this test a k-cross validation method was applied using k=5.

593 **5. Conclusions**

594 A novel method for protein characterization and classification has been proposed  
595 in this paper, which is inspired by and uses the algorithms from the facial recognition field.  
596 The first application of this method involves a challenging case of classification of highly  
597 homologous tubulin isotypes using as features some geometrical descriptors typically  
598 found within the context of face recognition analysis. While human faces and proteins  
599 represent very different biological structures, they are both free-form surfaces and the same  
600 types of geometrical features are adopted for their classification and recognition.

601 The aim of this study has been to implement different classifiers to be tested on the  
602 dataset previously built. In this work we used the following approaches: SVM with a linear  
603 RBF kernel, and a k-means algorithm. This methodology and the geometrical descriptors  
604 have been used for protein classification. The first classification was performed using the  
605 tubulin protein and 9 of its isotypes. The second application performed used two

606 structurally similar proteins: bovine tubulin and FtsZ and third application involved four  
607 unrelated proteins. In all cases very encouraging results were obtained.

608 It should be stressed that until now the use of RMSD as a measure of similarity has been  
609 prevalent in protein biophysics, especially regarding structural comparisons. However, this  
610 approach relies on a single number, which does not allow for feature extraction or more  
611 detailed shape comparisons, which the present methodology provides. A single parameter  
612 such as an RMSD value can answer the question if two proteins are structurally similar or  
613 not but does not address the issue regarding which features differ between them. For this  
614 reason, our method can assist in identifying structure-function dependence when  
615 comparing various proteins, even highly similar ones. Since we only investigate  
616 geometrical features, both physical and chemical properties are not directly involved in our  
617 method but can eventually be extracted by mapping geometrical features back onto to  
618 amino acid distributions underlying them. Also, the number of potential mutations of any  
619 protein, in particular tubulin, is astronomical. Consequently, brute force methods are not  
620 viable in classifying the role of specific mutations regarding the root causes of the  
621 conformational changes resulting in dysfunction of a given protein. However, our  
622 methodology based on machine learning approaches may offer a viable alternative with  
623 numerous potential applications in protein biophysics and beyond.

624 In this study, MD has been used to generate additional models of each protein for  
625 the training purpose where each of the models is extracted from equilibrated MD  
626 trajectories after clustering. Clustering of the trajectory provides us with different  
627 conformations of the same protein from MD trajectories. We used several snapshots from  
628 each structural cluster, which makes it possible to probe diverse sampling of the trajectory.

629 In future work, a larger set of protein structures will be used to address the issue of  
630 structural diversity across the entire PDB dataset consisting of over 150,000 entries.

631 The results obtained and reported here are significant: a 96.5 % accuracy for tubulin  
632 isotype classification, a 98.2 % accuracy for tubulin and FtsZ classification and a 98%  
633 accuracy for a set of four arbitrarily chosen protein structures. SVM is a classifier with  
634 competitive performance using a small dataset (< 3000 samples) and in this case the results  
635 are significant. The application of a neural network can be a future development using a  
636 convolutional type on a larger dataset (> 10,000 samples). The conclusion is that these  
637 geometrical descriptors work properly with the description of protein surfaces and they are  
638 accurate enough to properly describe protein surfaces.

639 Several future developments can be taken in consideration, namely:

- 640 • building a database adding more samples and more proteins;
- 641 • computing more features and testing classifiers, using more geometrical descriptors  
642 and filters;
- 643 • applying our method to different data set for the purpose of protein classification  
644 such as Hemoglobin classification [reference: Clang et al. ]
- 645 • developing more data augmentation techniques to enlarge the dataset;
- 646 • identifying specific important features on a protein, for example a binding pocket  
647 for a ligand or a protein-protein interaction interface.

648 Other important improvements will be performed in future tests. First, we will employ  
649 neural networks that were applied here with significant results with 3D geometrical  
650 descriptors [19]. Second, using a large dataset with unnecessarily numerous features the

651 classifier could be slow, so some feature optimization techniques will be implemented in  
652 order to [37] accelerate the training of the kernel machine.

653 **Author Contributions:** For research articles with several authors, a short paragraph  
654 specifying their individual contributions must be provided. The following statements  
655 should be used “conceptualization, F.M. and JAT; methodology, FM, JAT and LDG;  
656 software, LDG; validation, LDG; formal analysis, F.M. and JAT ; investigation, FM, JAT  
657 and LDG ; Matlab scripts, VR; resources, MA and EV; data curation, MA; writing—  
658 original draft preparation, LDG and MA; writing—review and editing, MA, F.M. and JAT;  
659 visualization, LDG and MA; supervision, F.M. and JAT; project administration, JAT;  
660 funding acquisition, JAT and EV”, please turn to the [CRediT taxonomy](#) for the term  
661 explanation.

662 **Funding:** This research received no external funding.

663 **Acknowledgments:** Computational infrastructure of the Pharma-matrix cluster at the  
664 Cross Cancer Institute is gratefully acknowledged.

665 **Conflicts of Interest:** The authors declare no conflict of interest.

666 **References:**

- 667 1. Gainza, P., et al., *Deciphering interaction fingerprints from protein molecular*  
668 *surfaces*. bioRxiv, 2019: p. 606202.
- 669 2. Planas-Iglesias, J., et al., *Understanding Protein–Protein Interactions Using Local*  
670 *Structural Features*. Journal of Molecular Biology, 2013. **425**(7): p. 1210-1224.
- 671 3. Rupp, B. and J. Wang, *Predictive models for protein crystallization*. Methods,  
672 2004. **34**(3): p. 390-407.

- 673 4. Saberi Fathi, S.M., D.T. White, and J.A. Tuszynski, *Geometrical comparison of*  
674 *two protein structures using Wigner-D functions: Geometrical Comparison of Protein*  
675 *Structures*. *Proteins: Structure, Function, and Bioinformatics*, 2014. **82**(10): p. 2756-2769.
- 676 5. Tsuda, K., H. Shin, and B. Schölkopf, *Fast protein classification with multiple*  
677 *networks*. *Bioinformatics*, 2005. **21**(suppl\_2): p. ii59-ii65.
- 678 6. Weston, J., et al., *Semi-supervised protein classification using cluster kernels*.  
679 *Bioinformatics*, 2005. **21**(15): p. 3241-3247.
- 680 7. Jain, P., J.M. Garibaldi, and J.D. Hirst, *Supervised machine learning algorithms for*  
681 *protein structure classification*. *Computational Biology and Chemistry*, 2009. **33**(3): p.  
682 216-223.
- 683 8. Masci, J., et al. *Geodesic Convolutional Neural Networks on Riemannian*  
684 *Manifolds*. 2015. IEEE Computer Society.
- 685 9. Monti, F., et al. *Geometric Deep Learning on Graphs and Manifolds Using Mixture*  
686 *Model CNNs*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern*  
687 *Recognition*. 2017.
- 688 10. Bronstein, M.M., et al., *Geometric Deep Learning: Going beyond Euclidean data*.  
689 *IEEE Signal Processing Magazine*, 2017. **34**(4): p. 18-42.
- 690 11. Espinosa, E., et al., *Classification of anticancer drugs—a new system based on*  
691 *therapeutic targets*. *Cancer Treatment Reviews*, 2003. **29**(6): p. 515-523.
- 692 12. Huang, C.-H., F.-L. Lee, and C.-J. Tai, *The  $\beta$ -tubulin gene as a molecular*  
693 *phylogenetic marker for classification and discrimination of the Saccharomyces sensu*  
694 *stricto complex*. *Antonie van Leeuwenhoek*, 2009. **95**(2): p. 135-142.

- 695 13. Ludueña, R.F., *Are tubulin isotypes functionally significant*. *Molecular Biology of*  
696 *the Cell*, 1993. **4**(5): p. 445-457.
- 697 14. Fitch, W.M., *Homology: a personal view on some of the problems*. *Trends in*  
698 *Genetics*, 2000. **16**(5): p. 227-231.
- 699 15. Richards, K.L., et al., *Structure–Function Relationships in Yeast Tubulins*.  
700 *Molecular Biology of the Cell*, 2000. **11**(5): p. 1887-1903.
- 701 16. Schlieper, D., et al., *Structure of bacterial tubulin BtubA/B: Evidence for horizontal*  
702 *gene transfer*. *Proceedings of the National Academy of Sciences of the United States of*  
703 *America*, 2005. **102**(26): p. 9170-9175.
- 704 17. Gunn, S.R., *Support Vector Machines for Classification and Regression*. p. 52.
- 705 18.
- 706 <https://u.pcloud.link/publink/show?code=XZwyRNkZdgxbscKvDcz9RcNn832cPYu>  
707 D3pRV)
- 708 19. Ciravegna, G., et al., *Assessing Discriminating Capability of Geometrical*  
709 *Descriptors for 3D Face Recognition by Using the GH-EXIN Neural Network*, in *Neural*  
710 *Approaches to Dynamics of Signal Exchanges*, A. Esposito, et al., Editors. 2020, Springer:  
711 Singapore. p. 223-233.
- 712 20. Cirrincione, G., et al., *Intelligent Quality Assessment of Geometrical Features for*  
713 *3D Face Recognition*, in *Neural Advances in Processing Nonlinear Dynamic Signals*, A.  
714 Esposito, et al., Editors. 2019, Springer International Publishing: Cham. p. 153-164.
- 715 21. Li, S.Z. and A.K. Jain, *Handbook of Face Recognition*. 2 ed. 2011, London:  
716 Springer-Verlag.

- 717 22. Marcolin, F., et al., *Three-dimensional face analysis via new geometrical*  
718 *descriptors*, in *Advances on Mechanics, Design Engineering and Manufacturing* :  
719 *Proceedings of the International Joint Conference on Mechanics, Design Engineering &*  
720 *Advanced Manufacturing (JCM 2016), 14-16 September, 2016, Catania, Italy*, B. Eynard,  
721 et al., Editors. 2017, Springer International Publishing: Cham. p. 747-756.
- 722 23. Marcolin, F. and E. Vezzetti, *Novel descriptors for geometrical 3D face analysis*.  
723 *Multimedia Tools and Applications*, 2017. **76**(12): p. 13805-13834.
- 724 24. Koenderink, J. J. and Van Doorn, A. J., *Surface shape and curvature scales*. *Image and*  
725 *vision computing*, 1992, 10(8), 557-564.
- 726 25. Vezzetti, E. and F. Marcolin, *Geometrical descriptors for human face*  
727 *morphological analysis and recognition*. *Robotics and Autonomous Systems*, 2012. **60**(6):  
728 p. 928-939.
- 729 26. *MATLAB*. 2018, The MathWorks Inc.: Natick, Massachusetts.
- 730 27. *Anaconda Software Distribution*. *Computer software*. 2019, Anaconda,.
- 731 28. Van Rossum, G. and F.L. Drake Jr, *Python*. 2019, Centrum voor Wiskunde en  
732 Informatica Amsterdam, The Netherlands.
- 733 29. *Version 0.22.0 — scikit-learn 0.22 documentation* - [https://scikit-](https://scikit-learn.org/stable/whats_new/v0.22.html)  
734 [learn.org/stable/whats\\_new/v0.22.html](https://scikit-learn.org/stable/whats_new/v0.22.html).
- 735 30. *Download R-3.5.3 for Windows*. *The R-project for statistical computing* -  
736 <https://cran.r-project.org/bin/windows/base/old/3.5.3/>.
- 737 31. Löwe, J., et al., *Refined structure of  $\alpha\beta$ -tubulin at 3.5 Å resolution* | Edited by I. A.  
738 *Wilson*. *Journal of Molecular Biology*, 2001. **313**(5): p. 1045-1057.

- 739 32. *Molecular Operating Environment (MOE). Group, Chemical Computing.* 2012:  
740 Montreal, QC, Canada.
- 741 33. Oliva, M.A., S.C. Cordell, and J. Löwe, *Structural insights into FtsZ protofilament*  
742 *formation.* Nature Structural & Molecular Biology, 2004. **11**(12): p. 1243-1250.
- 743 34. D.A. Case, et al., *AMBER 2014.* 2014: University of California, San Francisco.
- 744 35. Konc, J., et al., *ProBiS-CHARMMing: Web Interface for Prediction and*  
745 *Optimization of Ligands in Protein Binding Sites.* Journal of Chemical Information and  
746 Modeling, 2015. **55**(11): p. 2308-2314.
- 747 36. Saberi Fathi, S.M. and J.A. Tuszynski, *A simple method for finding a protein's*  
748 *ligand-binding pockets.* BMC Structural Biology, 2014. **14**: p. 18.
- 749 37. Rahimi, A. and B. Recht, *Random Features for Large-Scale Kernel Machines.* p.  
750 10.

751 **Figures' captions:**

752 **Figure 1:** Structural similarities between tubulin and FtsZ proteins. The tubulin dimer  
753 consists of an  $\alpha$ -tubulin and a closely related  $\beta$ -tubulin monomer.  $\alpha\beta$ -tubulin heterodimers  
754 associate head to tail to form protofilaments and laterally to form the cylindrical MT wall.  
755 GTP and GDP nucleotides (ball and stick models) are bound to  $\alpha$  and  $\beta$  tubulin,  
756 respectively. (b) The FtsZ dimer consists of two identical monomers with GTP bound to  
757 N-terminals (blue). In both (a) and (b) N-terminals (blue) and C-terminals (red) are  
758 separated by H7 helices (green). N-terminal regions show the typical nucleotide-binding  
759 motif with parallel  $\beta$  sheets connected by  $\alpha$  helices known as the Rossmann fold. By  
760 comparing the two protein structures, the differences in C-terminal regions are obvious.  
761 GDP and GTP are shown in ball and stick models. The figures were rendered using the

762 MOE (Molecular Operating Environment) software. PDB ID for tubulin: 1JFF. PDB ID  
763 for FtsZ: 1W5B.

764 **Figure 2:** Valid solutions can be found with perceptron in a binary case(a) and the best  
765 theoretical solution that a SVM classifier can find (b).

766 **Figure 3:** Flow chart of the entire protein characterization and classification process.

767 **Figure 4:** Effects of applying different descriptors (a)  $F\_den2$  ,(b)  $sing$  (c) ,  $S_{mean}$  , (d)  
768  $k_{1mean}$  ,(e)  $k_{2median}$  , (f),  $g_{mean}$  ,and (g)  $H_{mean}$  to a human face (left column) and to the  
769 tubulin protein (right column)

770 **Figure 5:** Sequence alignment of  $\beta$  tubulin isotypes. Each of the human  $\beta$  tubulin isotypes  
771 that were identified in our screen of the UniProt databases were aligned using the MOE  
772 package. Prior to performing the alignment, the highly variable carboxy-terminal residues  
773 were removed from each sequence. This was done as the template structure, 1JFF, does not  
774 contain any of these residues. At each position within the alignment, dark blue boxes  
775 indicate identical residues; light blue boxes indicate residues that are conserved, while red  
776 boxes indicate residues that are divergent (poorly aligned).

777 **Figure 6:** a) Sequence similarity matrix and (b) sequence identity matrix of the studied  
778 tubulin isotypes. The matrices are heatmap color-coded (the darker the shade, the more  
779 similar the values are).

780 **Figure 7:** Tubulin protein image for two different rotations with respect to the Z-axis.  
781 The blue color-code represents not conserved and red color represents the more  
782 conserved as it shown in the scale bar. The images were taken from  
783 <https://probis.nih.gov/>.

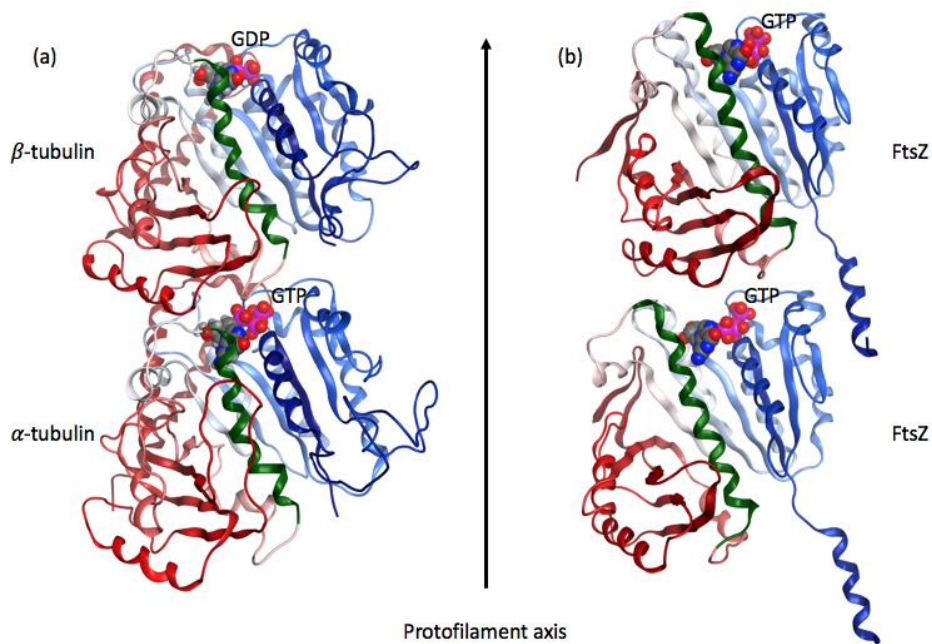
784 **Figure 8:** Protein data processing overview. The input consists of a 3D structure of a  
785 protein from either the PDB database or from homology modeling combined with MD  
786 simulations. The color selection in the input structure is arbitrarily chosen for better  
787 visualization. The output consists of geometrical descriptor values obtained from a facial  
788 recognition algorithm.

789 **Figure 9:** 9 bin histograms calculated using (a)  $F_{den2}$ , (b)  $g_{mean}$  and (c)  $H_{mean}$   
790 geometrical descriptor

791 **Figure 10:** Confusion matrix of SVM classifier using the RBF kernel.

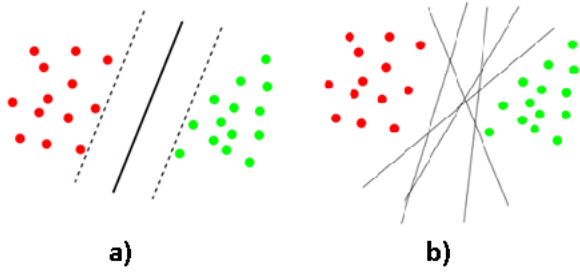
792 **Figures:**

793 **Figure 1**



794

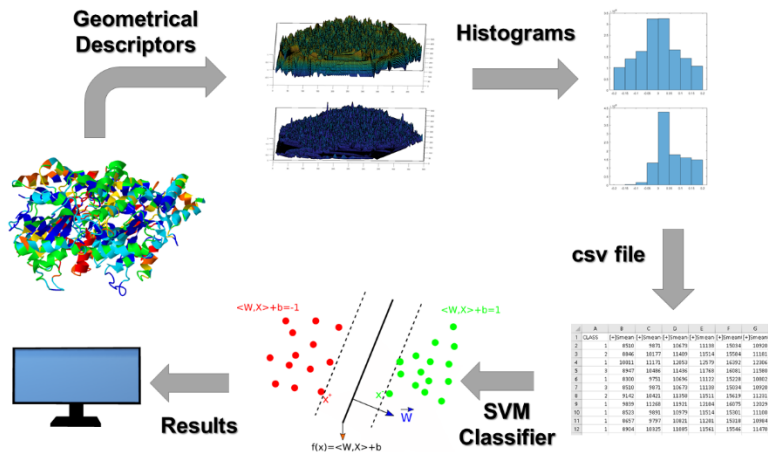
795 **Figure 2**



796

797

Figure 3



798

799

800

801

802

803

804

805

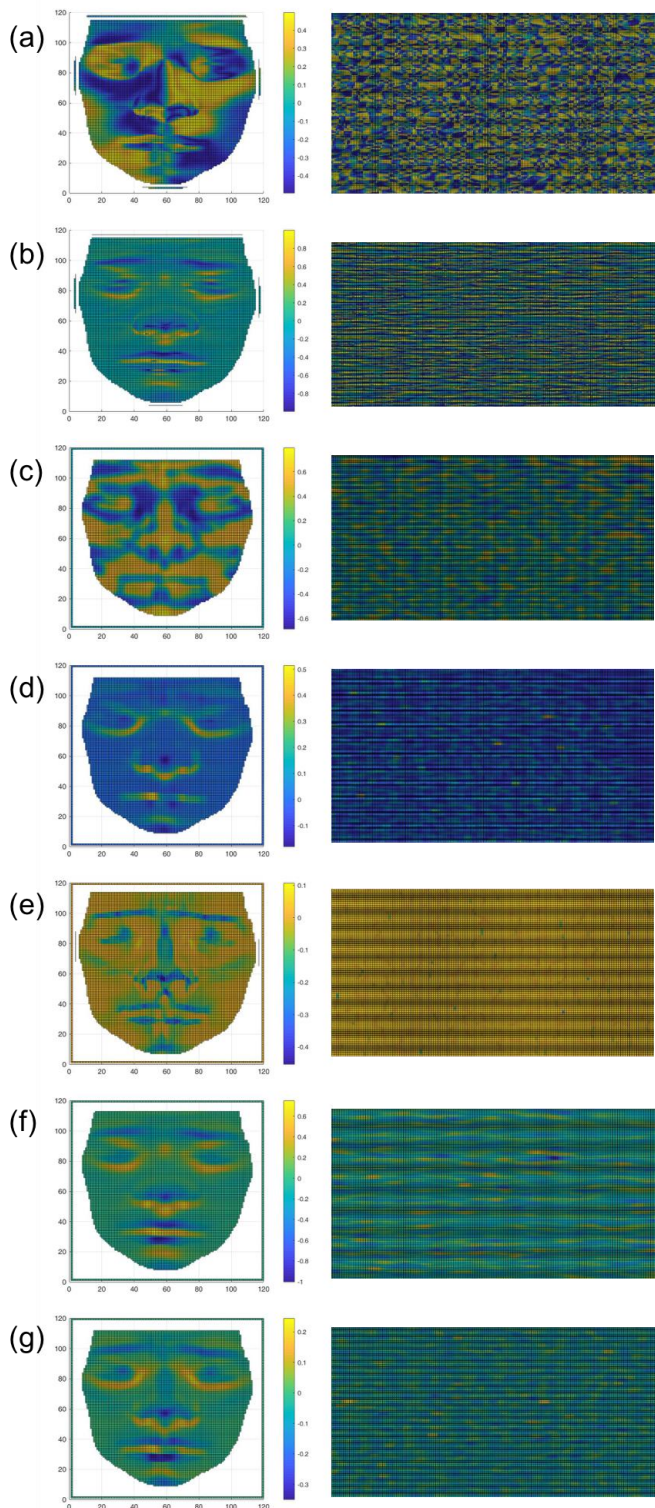
806

807

808

	A	B	C	G	E	D	Q
1	CLASS	(-)-Sinepin	(-)-Sinepin	(-)-Sinepin	(-)-Sinepin	(-)-Sinepin	(-)-Sinepin
2	1	8900	9679	10479	11179	11914	12629
3	2	8900	9677	10480	11154	11904	12611
4	1	10911	11175	11811	12779	14102	15206
5	2	8907	9490	10400	11796	12911	13988
6	1	8900	9770	10436	11179	11739	12601
7	2	8900	9673	10479	11179	11914	12611
8	2	11410	11843	12799	13911	15019	16111
9	1	9699	11100	11911	12844	13979	15119
10	1	8511	9091	10879	11914	13101	14309
11	1	1017	11917	13011	14101	15111	16111
12	1	8914	10101	11011	11914	13011	14179

809 **Figure 4**



810

811

812 **Figure 5**

Chain	1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75																																																						
1: beta1	M	R	E	I	V	H	I	Q	A	G	Q	C	N	I	G	A	K	F	W	E	V	I	S	D	E	H	G	I	D	P	T	G	T	Y	H	G	S	D	L	Q	L	D	R	I	S	V	Y	N	E	A	T	G	G	K	Y	V	P	R	A	I	L	V	D	L	E	P	G	T	M	S
2: beta1IA	M	R	E	I	V	H	I	Q	A	G	Q	C	N	I	G	A	K	F	W	E	V	I	S	D	E	H	G	I	D	P	T	G	S	Y	H	G	S	D	L	Q	L	E	R	I	N	V	Y	N	E	A	A	G	N	K	Y	V	P	R	A	I	L	V	D	L	E	P	G	T	M	S
3: beta1IB	M	R	E	I	V	H	I	Q	A	G	Q	C	N	I	G	A	K	F	W	E	V	I	S	D	E	H	G	I	D	P	T	G	S	Y	H	G	S	D	L	Q	L	E	R	I	N	V	Y	N	E	A	T	G	N	K	Y	V	P	R	A	I	L	V	D	L	E	P	G	T	M	S
4: beta1II	M	R	E	I	V	H	I	Q	A	G	Q	C	N	I	G	A	K	F	W	E	V	I	S	D	E	H	G	I	D	P	S	G	N	Y	V	G	S	D	L	Q	L	E	R	I	S	V	Y	N	E	A	S	H	K	Y	V	P	R	A	I	L	V	D	L	E	P	G	T	M	S	
5: beta1VA	M	R	E	I	V	H	L	Q	A	G	Q	C	N	I	G	A	K	F	W	E	V	I	S	D	E	H	G	I	D	P	T	G	T	Y	H	G	S	D	L	Q	L	E	R	I	N	V	Y	N	E	A	T	G	G	N	Y	V	P	R	A	V	L	V	D	L	E	P	G	T	M	S
6: beta1VB	M	R	E	I	V	H	L	Q	A	G	Q	C	N	I	G	A	K	F	W	E	V	I	S	D	E	H	G	I	D	P	T	G	T	Y	H	G	S	D	L	Q	L	E	R	I	N	V	Y	N	E	A	T	G	G	K	Y	V	P	R	A	V	L	V	D	L	E	P	G	T	M	S
7: betaV	M	R	E	I	V	H	I	Q	A	G	Q	C	N	I	G	T	K	F	W	E	V	I	S	D	E	H	G	I	D	P	A	G	G	V	V	G	S	A	L	Q	L	E	R	I	N	V	Y	N	E	S	S	Q	K	Y	V	P	R	A	A	L	V	D	L	E	P	G	T	M	S	
8: betaVI	M	R	E	I	V	H	I	Q	I	G	Q	C	N	I	G	A	K	F	W	E	M	I	G	E	H	G	I	D	L	A	G	S	D	R	G	A	S	A	L	Q	L	E	R	I	S	V	Y	N	E	A	Y	G	R	K	Y	V	P	R	A	V	L	V	D	L	E	P	G	T	M	S
9: betaVIII	M	R	E	I	V	L	T	Q	I	G	Q	C	N	I	G	A	K	F	W	E	I	S	D	E	H	A	I	D	S	A	G	T	Y	H	G	S	H	L	Q	L	E	R	I	N	V	Y	N	E	A	S	G	G	R	Y	V	P	R	A	V	L	V	D	L	E	P	G	T	M	S	

Chain	76	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150																																																								
1: beta1	V	R	S	G	P	F	G	Q	I	F	R	P	D	N	F	V	F	G	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	S	V	L	D	V	V	R	K	E	A	E	S	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L	
2: beta1IA	V	R	S	G	P	F	G	Q	I	F	R	P	D	N	F	V	F	G	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	S	V	L	D	V	V	R	K	E	S	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L			
3: beta1IB	V	R	S	G	P	F	G	Q	I	F	R	P	D	N	F	V	F	G	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	S	V	L	D	V	V	R	K	E	S	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L			
4: beta1II	V	R	S	G	A	F	G	H	L	F	R	P	D	N	F	I	F	G	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	S	V	L	D	V	V	R	K	E	C	E	N	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L	
5: beta1VA	V	R	S	G	P	F	G	Q	I	F	R	P	D	N	F	V	F	G	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	D	A	V	L	D	V	V	R	K	E	A	E	S	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L
6: beta1VB	V	R	S	G	P	F	G	Q	I	F	R	P	D	N	F	V	F	G	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	S	V	L	D	V	V	R	K	E	A	E	S	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L	
7: betaV	V	R	S	G	P	F	G	Q	I	F	R	P	D	N	F	I	F	G	T	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	V	D	A	V	L	D	V	V	R	K	E	C	H	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L	
8: betaVI	I	R	S	K	L	G	A	L	F	Q	P	D	S	F	V	H	G	N	S	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	I	N	V	L	E	V	V	R	H	E	S	C	D	C	L	Q	G	F	I	V	H	S	L	G	G	G	T	S	G	M	G	T	L			
9: betaVIII	V	R	S	G	P	F	G	Q	V	F	R	P	D	N	F	I	F	G	C	G	A	G	N	N	W	A	K	G	H	Y	T	E	G	A	E	L	M	E	S	V	M	D	V	V	R	K	E	A	E	S	C	D	C	L	Q	G	F	L	T	H	S	L	G	G	G	T	S	G	M	G	T	L

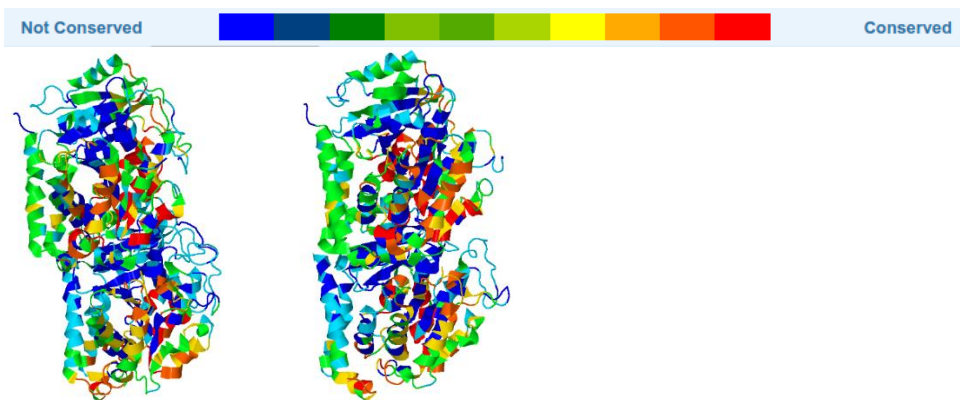
Chain	151	155	160	165	170	175	180	185	190	195	200	205	210	215	220	225																																																									
1: beta1	L	I	S	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	V	H	Q	L	V	E	N	T	D	E	T	Y	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L	
2: beta1IA	L	I	S	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	V	H	Q	L	V	E	N	T	D	E	T	Y	S	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L	
3: beta1IB	L	I	S	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	V	H	Q	L	V	E	N	T	D	E	T	Y	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L	
4: beta1II	L	I	S	K	V	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	I	H	Q	L	V	E	N	T	D	E	T	Y	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	A	P	T	T	Y	G	D	L
5: beta1VA	L	I	S	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	V	H	Q	L	V	E	N	T	D	E	T	Y	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L	
6: beta1VB	L	I	S	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	V	H	Q	L	V	E	N	T	D	E	T	Y	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L	
7: betaV	L	I	S	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	T	L	S	V	H	Q	L	V	E	N	T	D	E	T	Y	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L	
8: betaVI	L	M	N	K	I	R	E	E	P	D	R	I	M	N	T	F	S	V	M	P	S	P	K	V	S	D	T	V	V	E	P	Y	N	A	V	L	S	I	H	Q	L	I	E	N	A	D	A	C	F	C	I	D	N	E	A	L	Y	D	I	C	F	R	T	L	K	L	T	P	T	Y	G	D	L
9: betaVIII	L	L	S	K	I	R	E	E	P	D	R	I	I	N	T	F	S	L	P	S	P	K	V																																																		

815 **Figure 6**

(a)	1	2	3	4	5	6	7	8	9	(b)	1	2	3	4	5	6	7	8	9
1:betaI		99.1	99.5	97.0	99.5	99.8	96.3	90.4	96.0	1:betaI		97.0	97.4	93.9	97.4	98.4	92.5	80.1	89.7
2:betaIIA	99.1		99.5	97.4	99.1	99.3	96.3	90.4	95.8	2:betaIIA	97.0		99.5	93.4	96.3	97.7	92.5	80.8	89.9
3:betaIIB	99.5	99.5		97.7	99.5	99.8	96.5	90.6	96.0	3:betaIIB	97.4	99.5		93.7	96.7	98.1	92.7	81.0	90.2
4:betaIII	97.0	97.4	97.7		97.2	97.2	97.9	90.9	93.7	4:betaIII	93.9	93.4	93.7		93.2	93.9	94.4	80.1	87.6
5:betaIVA	99.5	99.1	99.5	97.2		99.8	96.5	90.4	95.8	5:betaIVA	97.4	96.3	96.7	93.2		98.6	93.4	80.3	90.2
6:betaIVB	99.8	99.3	99.8	97.2	99.8		96.5	90.6	96.3	6:betaIVB	98.4	97.7	98.1	93.9	98.6		92.7	80.3	91.1
7:betaV	96.3	96.3	96.5	97.9	96.5	96.5		90.2	93.2	7:betaV	92.5	92.5	92.7	94.4	93.4	92.7		80.1	87.1
8:betaVI	90.4	90.4	90.6	90.9	90.4	90.6	90.2		89.2	8:betaVI	80.1	80.8	81.0	80.1	80.3	80.3	80.1		78.2
9:betaVIII	96.0	95.8	96.0	93.7	95.8	96.3	93.2	89.2		9:betaVIII	89.7	89.9	90.2	87.6	90.2	91.1	87.1	78.2	

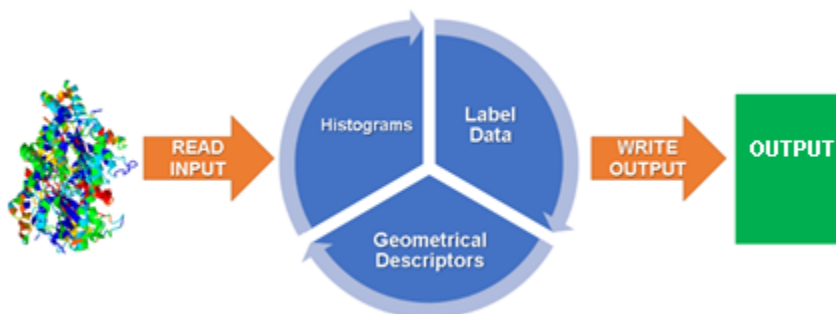
816  
817

818 **Figure 7**



819

820 **Figure 8**



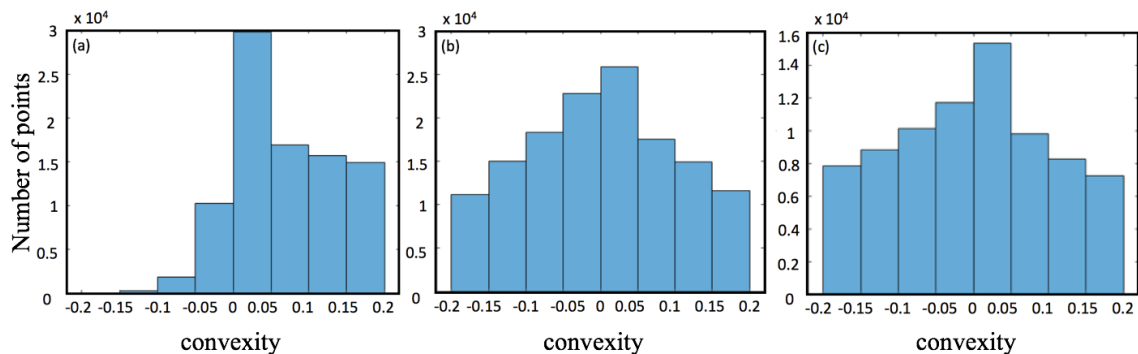
821

822

823

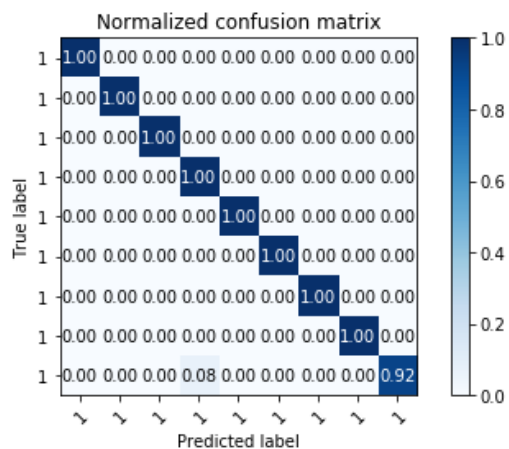
824

825 **Figure 9**



826  
827

828 **Figure 10.**



829

830 **Tables' captions:**

831 **Table 1:** Numbers of tubulin isotype structures used.

832 **Table 2:** Sample numbers in the binary classification between tubulin and FtsZ.

833 **Table 3:** Number of Tubulin isotypes used.

834 **Table 4:** Tubulin isotypes accuracy results.

835 **Table 5:** Accuracy results for the tubulin and FtsZ binary classification.

836 **Table 6:** 2R6R, 2VAM, 2VAP and 2VAM samples.

837 **Table 7:** 2R6R, 2VAM, 2VAP and 2VAM experiment.

838 **Tables:**

839 **Table 1**

<b>Isotypes</b>	Beta I	Beta IIa	Beta IIb	Beta III	Beta IVa	Beta IVb	Beta V	Beta VI	Beta VIII
<b>Samples</b>	123	128	94	57	128	68	107	62	125

840

841 **Table 2**

<b>Protein</b>	<b>Samples</b>
Tubulin	112
FtsZ	65

842

843 **Table 3**

<b>Isotypes</b>	Beta I	Beta IIa	Beta IIb	Beta III	Beta IVa	Beta IVb	Beta V	Beta VI	Beta VIII
<b>Samples</b>	1	2	3	4	5	6	7	8	9

844

845 **Table 4**

<b>Classifier</b>	<b>Accuracy</b>
SVM with RBF kernel	96.5 %
SVM with linear kernel	92.4 %
k-means	76.6 %

846

847 **Table 5**

<b>Classifier</b>	<b>Accuracy</b>
SVM with RBF kernel	98.2 %
SVM with linear kernel	97.0 %
k-means	72.3 %

848

849 **Table 6**

<b>Proteins</b>	2R6R	2VAW	2VAP	2VAM
<b>Samples</b>	175	170	168	170

850

851 **Table 7**

852

Classifier	Accuracy
SVM with RBF kernel	97.1 %
SVM with linear kernel	98.0 %
k-means	62.3 %