Building blocks for commodity augmented reality-based molecular visualization and modeling in web browsers

Luciano Andrés Abriata[1]

[1] École Polytechnique Fédérale de Lausanne and Swiss Institute of Bioinformatics, Lausanne, Switzerland.

Corresponding Author:
Luciano A. Abriata[1]
AAB044 Station 19, CH-1015 Lausanne, Switzerland
Email address: luciano.abriata@epfl.ch

Abstract

For years, immersive interfaces using virtual and augmented reality (AR) for molecular visualization and modeling have promised a revolution in the way how we teach, learn, communicate and work in chemistry, structural biology and related areas. However, most tools available today for immersive modeling require specialized hardware and software, and are costly and cumbersome to setup. These limitations prevent wide use of immersive technologies in education and research centers in a standardized form, which in turn prevents large-scale testing of the actual effects of such technologies on learning and thinking processes. Here, I discuss building blocks for creating marker-based AR applications that run as web pages on regular computers, and explore how they can be exploited to develop web content for handling virtual molecular systems in commodity AR with no more than a webcam- and internet-enabled computer. Examples span from displaying molecules, electron microscopy maps and molecular orbitals with minimal amounts of HTML code, to incorporation of molecular mechanics, real-time estimation of experimental observables and other interactive resources using JavaScript. These web apps provide virtual alternatives to physical, plastic-made molecular modeling kits, where the computer augments the experience with information about spatial interactions, reactivity, energetics, etc. The ideas and prototypes introduced here should serve as starting points for building active content that everybody can utilize online at minimal cost, providing novel interactive pedagogic material in such an open way that it could enable mass-testing of the effect of immersive technologies on chemistry education.

Introduction

For a long time it has been suggested that visual immersive analytics based in virtual reality (VR), augmented reality (AR) and other advanced forms of human-computer interactions have enormous potential in assisting thinking processes in scientific research and in education, especially in areas

40 of science that deal with abstract objects, objects much smaller or larger than human dimensions,

41 objects that are hard to acquire and handle due to high costs, scarcity or fragility, and very large

42 amounts of data (O'Donoghue et al., 2010; Matthews, 2018; M. Krichenbauer et al., 2018; Sommer

43 et al., 2018). Chemistry and structural biology are examples of such disciplines where AR and VR

44 have been attributed high potential in education and research, by providing hybrid

45 physical/computational interfaces to handle and explore virtual molecules in real 3D space

46 augmented with real-time overlay of information from databases and calculations. However, the

47 actual impact of immersive technologies on teaching, learning and working in chemistry still

48 requires deep evaluation (M. Fjeld & B. M. Voegtli, 2002; Pence, Williams & Belford, 2015;

49 Matthews, 2018; Bach et al., 2018; Yang, Mei & Yue, 2018). Such evaluation has progressed very

50 slowly due to the complex software setups and the specialized hardware needed, which limit

51 availability, reach, adoption, and thus testing. Indeed this limitation is shared more broadly with

52 other potential applications of AR and VR in science, which so far "[…] remain niche tools for

53 scientific research" (Matthews, 2018).

54   In the last decade several groups have been studying ways to achieve immersive

55 environments for chemistry and structural biology using VR and AR (Gillet et al., 2004, 2005;

56 Maier, Tönnis & GudrunKlinker, 2009; Maier & Klinker, 2013; Hirst, Glowacki & Baaden, 2014;

57 Berry & Board, 2014; Martínez-Hung, García-López & Escalona-Arranz, 2016; Vega Garzón,

58 Magrini & Galembeck, 2017; Balo, Wang & Ernst, 2017; Goddard et al., 2018b; Wolle, Müller &

59 Rauh, 2018; O'Connor et al., 2018; Ratamero et al., 2018; Müller et al., 2018; Goddard et al.,

60 2018a; Stone, 2019). Such interfaces allow handling molecules over 6 degrees of freedom and

61 with both hands, in immersive 3D. They are expected to overcome the limitations of traditional

62 software based on screen, mouse and keyboard, thus enabling more intuitive, fluid exploration of

63 molecular features and data. At the time of writing, most of these works are not complete AR or

64 VR versions of fully-fledged programs, but rather prototypes, proofs of concept and case studies

65 on how humans interact with virtual molecules in AR or VR. Notable highlights moving towards

66 complete immersive molecular visualization and modeling programs are the new rewrite of

67 Chimera, ChimeraX, which was optimized for the new GPUs and incorporates support for VR

68 experiences (Goddard et al., 2018b), VRmol (Xu et al., 2019), new VR plugins for the VMD

69 molecular graphics program (Stone, 2019), and a few commercial programs like Autodesk's

70   Molecule   Viewer   (https://autodeskresearch.com/groups/lifesciences)   or   Nanome

71   (https://nanome.ai/), all with interfaces specifically tailored for VR.

72   Most of these works suffer from two severe limitations. First, all but a few exceptions

73   require hardware such as head-mount displays (helmets, headsets or goggles like MS Hololens,

74   Oculus Rift, HTC Vibe, etc.) or immersive installations with large surround screens plus 3D-

75   handheld input devices and the corresponding computer and GPU hardware. The few remarkable

76   exceptions are prototypes using ordinary webcam-enabled smartphones (Balo, Wang & Ernst,

77   2017) or laptops (Gillet et al., 2004). The second limitation is that such specialized hardware

78   requires specialized programs that often must be correctly interfaced to couple the different

79   components required for an AR or VR experience, i.e. tracking limbs, handheld devices or AR

80   markers, then running calculations on molecular data, and finally displaying results and molecular

81   graphics on screen (for examples see (Ratamero et al., 2018) and (Gillet et al., 2005)). Some of

82   these programs are only compatible with specific VR devices and many are not free software.

83   Overall, then, despite the dropping costs, access to these tools still requires investment in the order

84   of hundreds to low-thousand American dollars per user, and software interfacing that may not be

85   available to lay students and teachers. It is therefore unlikely VR will achieve the ideal of "one

86   device per student" within the next few years. To date, these solutions are not widely used across

87   the world, and their costs make them totally out of reach for educational centers in developing

88   countries. Additionally, most current setups are limited to VR, but it has been shown that AR is

89   more suited for educational purposes because by not occluding the view of the user's own limbs,

90   it results in better motion coordination and object handling than VR (Sekhavat & Zarei, 2016; Y.

91   Gaffary et al., 2017; M. Krichenbauer et al., 2018). Furthermore, in AR the view of the world is

92   not obstructed thus allowing students and teachers to interact more easily.

93   The purpose of this work is to demonstrate that client-side web technologies have matured

94   enough to enable web pages for AR-based molecular visualization and modeling running just on

95   web browsers in regular webcam-equipped computers. This enables the easy creation of immersive

96   educational material that can be employed by students and teachers at very affordable costs and

97   with very simple setups. All they must do is access a webpage, enable webcam activity in the

98   browser, and show to the webcam a printed AR marker on which the molecules will be displayed.

99   From the developer side, the code is made simple thanks to a large number of libraries; in fact

100   visualization-only applications are achievable just with HTML code while interactivity can be

101   incorporated through custom JavaScript.

102   This article is organized in two parts. Part 1 provides a practical overview of the main

103   building blocks available as of 2019 to program AR apps in web pages, with a focus on ways to

104   achieve molecular visualization and modeling. It also briefly explores ways to handle gesture- and

105   speech-based commands, molecular mechanics, calculation of experimental observables,

106   concurrent collaboration through the world wide web, and other human-computer interaction

107   technologies available in web browsers. Part 2 of the article showcases prototype web apps for

108   specific tasks of practical utility in pedagogical and research settings. These web apps are based

109   on open, commodity technology that requires only a modern browser "out of the box", so

110   educators, students and researchers are free to try out all these examples on their computers right

111   away by following the provided links.

112

Part 1: Overview of building blocks for immersive molecular modeling in web browsers

Virtual and augmented reality

116   At the core of immersive experiences are visualizations based on virtual or augmented

117   reality methods. While VR is probably best experienced with VR goggles to suppress any side

118   view of the real world, AR is more amenable to devices like desktop or laptop computers, tablets

119   and smartphones, making it better suited for commodity solutions, and has the additional

120   advantages outlined in the introduction. The methods presented in this article make use of marker-

121   based AR in a web-based setup that functions as an "augmented reality mirror" where the user sees

122   him/herself with the virtual objects, here molecules, overlaid on the markers he/she holds in their

123   hands (Figure 1A,B). The following descriptions focus on this technology and how it can be

124   implemented in web apps by using HTML, CSS and JavaScript code as standard web pages that

125   are hosted at servers but run entirely on local computers (Figure 1C).

126   The WebGL API provides powerful 2D and 3D graphing capabilities using GPU resources,

127   in a format fully integrable with other HTML elements, APIs and JavaScript libraries, without the

128   need of plug-ins; and is highly standardized across browsers. It is thus possible to couple all

129   elements required to build up an AR experience directly inside the web page source code. A

130   handful of JavaScript libraries exploit WebGL to facilitate rendering of 3D scenes, Three.js

131 (https://threejs.org/) being probably the most widely used. In turn, tools like A-Frame

132 (https://aframe.io/) provide entity component system frameworks that wrap Three.js into HTML

133 language tags through polyfills, greatly facilitating the development of AR and VR scenes. The

134 examples presented in Part 2 showcase either direct use of Three.js or of Three.js through A-Frame

135 for AR. These libraries/frameworks can be used either by (i) loading pre-made models of the

136 molecular systems in formats like Wavefront's OBJ+MTL files exported straight out of molecular

137 visualization programs like VMD (Humphrey, Dalke & Schulten, 1996), UnityMol (S.

138 Doutreligne et al., 2014; Wiebrands et al., 2018), ChimeraX (Goddard et al., 2018b), etc., possibly

139 further edited with a program like Blender as in (Martínez-Hung, García-López & Escalona-

140 Arranz, 2016)); or (ii) employing 3D primitives (like spheres, cylinders, etc.) to draw the molecular

141 systems from scratch with their atomic coordinates. Use of Wavefront models is much simpler

142 (only a few lines of HTML code to load and display objects) and allows seamless display of any

143 element exported from molecular graphics programs, including not only different representations

144 of molecules but also isosurfaces as those needed to visualize volumetric data describing molecular

145 orbitals, electron microscopy maps, etc. On the other hand, using 3D primitives requires larger

146 pieces of code to create all 3D objects from atomic coordinates, but in turn allows for fine dynamic

147 control of shapes, sizes, colors, and positions, which are key to interactivity. Another downside of

148 Wavefront models is that files can become quite large for high-resolution textures.

149

150 **Object detection and tracking**

151 The other key component required for AR and VR is a means to detect and track objects or parts

152 of the user's body such as his/her hands, in order to manipulate virtual objects. Applications using

153 *ad hoc* hardware use sensors and cameras that track the user's position in space and handheld

154 controllers that the user sees as virtual tweezers or hands, to directly move objects in space. For

155 commodity AR/VR in web browsers, solutions rely on JavaScript versions of tracking libraries

156 that implement computer vision algorithms through the webcam feed, like ARToolKit (Kato,

157 1999) among other similar solutions. These libraries track user-defined 2D markers (like those in

158 Figure 1A, Figure S1, Figure S2) in space and make their computed coordinates available to the

159 graphics algorithms. Two especially interesting libraries, used in many of the examples presented

160 in Part 2, are AR.js (https://github.com/jeromeetienne/AR.js) and its A-Frame wrap

161 (https://jeromeetienne.github.io/AR.js/aframe/). These enable highly simplified AR/VR, even

162 using exclusively HTML code for simple developments.

163 It is important to note that in marker-based AR different viewers looking at the same

164 physical marker receive different perspectives of it and hence of the rendered virtual object, just

165 as if it was a real object in real space (Figure S3). This easily enables multi-user AR in a common

166 room, as would be useful in a classroom setting where students and teachers look at the same

167 virtual molecule.

168 An alternative to traditional marker-based AR should in principle be possible by using a

169 plain hand tracking JavaScript library like Handtracking.js. Another slightly more expensive

170 approach but possibly better in tracking performance is using a device like the infrared-based Leap

171 Motion Controller (https://www.leapmotion.com/), which includes a JavaScript library to feed

172 positional information from the device into the web app. Unfortunately, however, there are

173 currently no ready-to-use libraries that couple these input tools to WebGL graphics, so their use

174 would require further developments.

175 Current JavaScript libraries for computer vision allow even more complex object tracking.

176 One interesting example is gesture recognition by the WebGazer.js library, which analyzes face

177 features to estimate where on the screen the user is looking at (Papoutsaki et al., 2016). In

178 molecular visualization, this can be used for example to automatically rotate regions of interest to

179 the front of the field of view, as shown in Figure S4.

180

181 **Speech-based interfaces**

182 A speech-based interface can be very useful for situations in which the user's hands are

183 busy holding objects, as in AR/VR applications. In-browser speech recognition APIs enable

184 implementation of speech recognition very easily, especially through libraries like Annyang (Ater,

185 2019) which is used in some of the examples of this article. These libraries usually allow working

186 in two modes, one where the browser waits for specific commands (while accepting variable

187 arguments) and one where the browser collects large amounts of free text that are then made

188 available to the environment. The former allows direct activation of functions without the need for

189 the user to click on the screen. The second option opens up the possibility of automatically

190 detecting subjects, actions and concepts that are fed to artificial intelligence, or just predefined

191 rules, that the computer will analyze in background. For example, when two users are discussing

192 the interaction surface between two proteins and mention certain residues, the computer could
193 automatically mine NIH's PubMed repository of papers for mentions of said residues. This may
194 seem far-fetched, but is essentially the same technology that underlies automatic advertising and
195 suggestions based on users' various inputs and usage statistics in software and websites. The
196 problem of intelligently suggesting chemical and biological information related to a topic or object
197 has already been addressed for some time, for example in information augmentation tools like
198 Reflect (Pafilis et al., 2009) and advanced text-mining tools (Rebholz-Schuhmann, Oellrich &
199 Hoehndorf, 2012). The evolution of science-related standards are very important in this regard,
200 formats and content for the semantic web (Hendler, 2003) and of machine-readable scientific
201 databases and ontologies that standardize knowledge and data.

202

203 **Intensive calculations**

204 As recently reviewed in a dedicated issue of *Computing and Science in Engineering* (M. DiPierro,
205 2018), JavaScript has become very powerful by including language subsets specialized for speed,
206 optimized just-in-time compilation, methods to program background scripts, and libraries to
207 perform multi-core and on-GPUs computing to accelerate intensive calculations. It is in fact now
208 possible to transpile from C/C++ and other languages directly to JavaScript retaining close-to-
209 native execution speeds, allowing web browsers to support quite complex data analysis and
210 visualization directly in web pages (Abriata et al., 2017; Abriata, 2017a). This opens up the
211 possibility of simulating molecular mechanics and experimental data, performing numerical data
212 analysis and even handling data in neural networks, directly inside the molecular modeling web
213 app to enable real-time feedback as the user manipulates the molecular systems. Some of the
214 prototypes presented in Part 2 include such examples.

215 Rather than manually implementing calculations, a number of libraries are now available
216 for specific applications that help to save code writing time and bring the additional advantage of
217 being developed by specialists (Abriata, 2017a). One particularly useful library in the scope of this
218 paper is Cannon.js (https://schteppe.github.io/cannon.js/), an engine for simulating rigid body
219 mechanics that integrates with Three.js and A-Frame. These engines use numerical methods to
220 propagate motions of solid bodies connected by springs and other physical constraints in an
221 approximate but efficient way, thus adding realistic physics to the 3D bodies of AR and VR worlds.
222 Although these engines do not account for all the forces and phenomena of the atomic realm, such

223 as electrostatic interactions and quantum phenomena, they are useful in certain real scenarios of

224 molecular modeling. For example, the Integrative Modeling Platform software (used to put

225 together partial structures into bigger assemblies, driven by experimental data) includes one such

226 kind of physics engine for initial molecule-molecule docking (Russel et al., 2012). Furthermore,

227 implementation of more complex force fields is certainly possible, as exemplified by a JavaScript

228 transpilation of the OpenMD molecular dynamics engine (Jiang & Jin, 2017).

229

230 **Further building blocks**

231 Any other technology that facilitates interaction with the computer within a 3D environment, either

232 to deliver or obtain information, might be of use. For example, haptic feedback would be desirable

233 to confer a physical feel of interactions and clashes as in (Wollacott & Merz, 2007; Stocks,

234 Hayward & Laycock, 2009; Stocks, Laycock & Hayward, 2011; Matthews et al., 2019). Achieving

235 a good experience in haptic feedback currently requires specialized devices, and is an area of active

236 research (Bolopion et al., 2009, 2010), therefore it does not fit with the commodity hardware

237 criteria outlined in the introduction. Other rudimentary ways to achieve sensory feedback include

238 exploiting built-in vibration devices and touch-pressure sensing in touch screens, both handled by

239 JavaScript APIs.

240 Finally, a particularly interesting aspect of software running on web browsers is the ease

241 with which different users can connect to each other, just over the internet. Web apps can exploit

242 web sockets to achieve direct browser-to-browser links over which data can be transmitted freely,

243 with a server only intervening to establish the initial connection (V. Pimentel & B. G. Nickerson,

244 2012). For example, two or more users can concurrently work on a JSmol session by sharing just

245 mouse rotations and commands, appearing on all other users' screens with a minor delay (Abriata,

246 2017b). Such collaborative working technology could be adapted to complex immersive

247 environments to allow multiple users to work on chemical problems at a distance, essential for

248 scientific collaborations, demonstrations, and online teaching (Lee, Kim & Kang, 2012).

249

250 **Part 2: Prototype web apps showcasing example applications**

251 This section presents example AR web apps compatible with major web browsers in modern

252 computers, introducing features of increasing complexity. All examples were verified to run out

253 of the box in multiple web browsers on Windows, Linux and MacOS operating systems, in laptop

254 and desktop computers. All the examples are accessible through links in Table 1 and at

255 https://lucianoabriata.altervista.org/papersdata/tablepeerjcs2019.html, which further contains

256 links to demo videos. To run these examples the user needs to print the Hiro, Kanji or cube markers

257 as needed in each case (Figure 1A, Figure S1, Figure S2, and links on web pages). For simpler

258 handling, flat markers (Hiro and Kanji) may be glued on a flat surface mounted on a device that

259 can be easily rotated from the back, such as a small shaft perpendicular to the marker plane. The

260 cube marker is printed in a single page, folded and possibly glued to a solid cube made of wood,

261 plastic, rubber or similar material. Readers interested in the inner workings and in developing

262 content can inspect the source code of each webpage (Ctrl+U or Cmd+U in most browsers).

263 Several recommendations and basic troubleshooting for developers and users are given in Table

264 2.

265

266 **Introducing web browser-based AR for visualization**

267 The simplest way to achieve AR in web pages consists in displaying on the AR markers

268 representations exported from programs like VMD (Humphrey, Dalke & Schulten, 1996) in

269 Wavefront (OBJ+MTL) format. This can be achieved with a few lines of HTML code thanks to

270 libraries like AR.js for A-Frame, enabling the very easy creation of content for displaying any kind

271 of object handled by the exporting program. Figure 2A exemplifies this with a small molecule, 2-

272 bromo-butane, shown as balls and sticks. This small molecule is chiral at carbon 2; the Hiro marker

273 displays its R enantiomer while the Kanji marker displays the S enantiomer, both rendered from

274 the same pair of OBJ+MTL files but scaled as required for chirality inversion. Figure 2B shows

275 on the Hiro marker a protein complex rendered as cartoons with small molecule ligands rendered

276 as sticks (PDB ID 1VYQ, same example used by (Berry & Board, 2014)); and Figure 3 shows a

277 cartoon representation of a protein bound to a short segment of double stranded DNA rendered as

278 sticks (from PDB ID 1FJL) spinning on the Kanji marker. Figure 2D exemplifies display of VMD

279 isosurfaces to show a volumetric map of a bacteriophage attached to its host as determined through

280 electron microscopy (EMDB ID 9010). Two further examples feature combined representations

281 of atomic structure and volumetric data: Figure 2E shows a small peptide rendered as sticks and

282 docked inside the experimental electron map shown as a mesh (from PDB ID 3HYD), and Figure

283 2F shows the frontier molecular orbitals of BH_3 and NH_3 (Wavefront files kindly provided by G.

284 Frattini and Prof. D. Moreno, IQUIR, Argentina).

285   The next level of complexity is building up scenes from 3D primitives, which brings the

286   advantage over ready-made models that all the elements can be handled independently, thus

287   allowing the possibility of incorporating interactivity. This can be achieved either through A-

288   Frame with AR.js, thus requiring only HTML code for display as in the Wavefront-based examples

289   above, or through libraries that require additional JavaScript code but in turn enable more complex

290   functionalities. Exemplifying the former case, Figure 2G uses A-Frame spheres and cylinders

291   placed at coordinates computed from atomic positions, colored by atom type and assigned atom-

292   specific radii, to render a model of a small protein helix on the Kanji marker. The latter case of

293   using   pure   JavaScript   is   illustrated   by   the   jsartoolkit   library

294   (https://github.com/artoolkit/jsartoolkit5), in an example where six markers arranged on the faces

295   of a cube are coupled to drive a single object in full 3D. Lastly, combining this with direct use of

296   Three.js to draw 3D primitives, the web app in Figure 2H allows visualization and manipulation

297   of any molecule loaded in PDB format in full 3D, in this case (D)-glucopyranose.

298

299   **Adding interactivity: examples on small molecules**

300   Web apps using A-Frame can gain interactivity through portions of JavaScript code that read atom

301   coordinates and carry out calculations on them. Figure 3 shows another series of examples of

302   increasing complexity, focusing on small molecules. In Figure 3A, the user drives a lysine side

303   chain with the Hiro marker and a glutamate side chain with the Kanji marker. Each molecule is

304   anchored to the center of its corresponding AR marker through its alpha carbon atom. Their

305   protonation states correspond to neutral pH, so lysine is protonated hence its N atom (blue) is

306   charged by +1, whereas glutamate is deprotonated hence its O atoms (red) bear a total charge of -

307   1. Through simple JavaScript code the web app (i) represents with yellow dots the vector

308   connecting the lysine's N atom and one of the glutamate's O atoms; (ii) reports the distance

309   between these atoms and the corresponding attractive electrostatic force in real time; and (iii)

310   checks for and displays clashes between any pair of atoms of the two molecules. The code for (i)

311   is wrapped inside an auxiliary .js file, and the code for (ii) and (iii) is located between <script>

312   tags at the end of the HTML file. The distance, electrostatic and clash calculations are computed

313   inside a setInterval() function every 200 ms, and use the id attribute of the sphere tags to locate the

314   atomic coordinates. The distance calculation includes a correction for a zoom factor that scales

315   atom sizes and positions when the molecular coordinates are parsed into A-Frame HTML to

316   properly fit the screen. Clashes are detected when two spheres are within 3 Å and displayed as
317   semitransparent A-Frame spheres centered on the affected atoms.

318   A modified version of this example is also provided, incorporating a very simplistic
319   emulation of hydrogen bond detection and proton transfer (second link for Figure 3A in Table 1).
320   JavaScript code calculates and displays hydrogen bonds when the geometry permits, and randomly
321   "transfers" one proton from lysine's N atom to one of the O atoms of glutamate if the lysine is
322   protonated, or the other way around (actually spheres attached to each marker are hidden or shown
323   as needed to emulate proton transfer). Protons "jump" only when they are within 2 Å of the
324   receiving N or O atom; and they are set to jump back and forth to reflect 70% time-averaged
325   population of protonated lysine and 30% of protonated glutamate, to convey the feeling of different
326   acidic constants (10^{-pKa}) favoring protonation of the base. When 2 Å < N-O distance < 3 Å the web
327   app displays a yellow dotted line that represents a hydrogen bond between the potential receiver
328   heavy atom and the involved proton.

329   Similar emulation strategies could be easily used to build "interactive animations" for
330   exploring chemical and physical phenomena of much pedagogical use, as in the PhET interactive
331   simulations (Moore et al., 2014) but using AR to directly, almost tangibly, handle molecules. For
332   example, the app shown in Figure 3B illustrates stereoselectivity in the Diels-Alder reaction in
333   interactive 3D. This reaction occurs between a dienophile and a conjugated diene in a concerted
334   fashion, such that the side of the diene where the initial approach occurs defines the
335   stereochemistry of the product. The web app in this example allows users to visualize this in 3D
336   as they approach a molecule of 1,3-cyclohexadiene on the Hiro marker to a molecule of
337   chloroethene on the Kanji marker. As the two pairs of reacting C atoms approach each other, the
338   new bonds gain opacity until the product is formed. Additionally, the product formed in this
339   reaction is by itself an interesting molecule to visualize and manipulate in AR, because it contains
340   two fused six-membered rings which are hard to understand in 2D.

341   It should be noted that the examples provided here emulating reactivity are merely pictorial
342   visualizations of the mechanisms, and not based on any kind of quantum calculations. Such
343   calculations are too slow to be incorporated into immersive experiences where energies need to be
344   computed on the fly. However, novel machine learning methods that approximate quantum
345   calculations through orders-of-magnitude faster computations (like in (Smith, Isayev & Roitberg,
346   2017; Bartók et al., 2017; Paruzzo et al.)) could in the near future be coupled to AR/VR systems

347   to interactively explore reactivity in real time. Obviously, such tools could be useful not only for

348   education but also research, for example to interactively test the effect of chemical substituents on

349   a reaction, estimate effects on spectroscopic observables, probe effects of structural changes on

350   molecular orbitals, etc. It is already possible to integrate AR/VR with a physics engine, to add

351   realistic mechanics to the simulation. The web app in Figure 2H uses Cannon.js to simulate thermal

352   motions and thus give a sense of dynamics to the visualized system. In this web app, Cannon.js

353   handles the multi-atom system by treating atoms as spheres of defined radii connected by fixed-

354   distance constraints and whose velocities are continuously updated to match the set temperature,

355   leading to rotations around bonds. However, extension of Cannon.js to include additional force

356   field terms like dihedral angle terms and electrostatic interactions would be needed to achieve a

357   more complete and realistic modeling experience.

358

359   **AR-based modeling of biological macromolecules**

360   Figures 2 and 3 already include examples for visualizing certain biological macromolecules, and

361   the previous section introduced ways to incorporate interactivity into these web apps. This section

362   digs deeper into the development of interactive content more relevant to education and research in

363   structural biology, by exploring the incorporation of restraints, simple force fields and on-the-fly

364   simulation of experimental observables for biological macromolecules.

365   The example in Figure 3C uses JavaScript to calculate paramagnetic effects on the nuclear

366   magnetic resonance signal of a probe atom (black) attached to the Kanji AR marker, as it is moved

367   around the heme group of metmyoglobin attached to the Hiro AR marker. By applying standard

368   equations (Bertini, Turano & Vila, 1993) on the atomic coordinates, this web app simulates the

369   dipolar effects of the paramagnetic iron ion on the probe atom and displays the resulting

370   contribution   to   the   NMR   spectrum   using   the   Google   Charts   JavaScript   library

371   (https://developers.google.com/chart).

372   The web app shown in Figure 4A allows exploration of the interaction space of two proteins

373   that are known to form a complex in solution, specifically ubiquitin (red trace) and a ubiquitin-

374   interacting motif (UIM, blue trace) taken from PDB ID 2D3G (Hirano et al., 2006). The web app

375   simulates on-the-fly the small-angle X-ray scattering (SAXS) profiles expected from the relative

376   arrangement of the two proteins, and displays them overlaid onto an experimental profile in real

377   time as the user moves the proteins. This offers a way to interactively test compatibility of possible

378 docking poses with the experimental data. Although it cannot compete with the extensive sampling
379 achievable with molecular simulations, such an interactive tool could be useful for preliminary
380 analysis of SAXS data before starting complex calculations or to assist interpretation of the results
381 of such calculations. For simplicity and speed, in this example the SAXS profile calculation is
382 based on the Debye formula iterated through pairs of residues instead of through all pairs of atoms
383 as the full equation requires (Debye, 1915); however, realistic SAXS profiles of actual utility in
384 modeling can be achieved with coarse graining strategies and proper parameterization of the
385 scattering centers (Stovgaard et al., 2010). This web app further includes a rudimentary residue-
386 grained force field (i.e. describing each amino acid with one backbone and one side chain bead) to
387 detect clashes, and a predefined binding coordinate which upon activation brings the two
388 molecules together. Activation of SAXS profile simulation, clash-detecting force field and binding
389 coordinate are controlled by voice commands, required because the user's hands are busy handing
390 the markers. This proceeds through the browser's cloud-based speech recognition API so does not
391 consume much resources. The successful combination of all these different elements (AR, 3D
392 visualization, calculations, and speech recognition) illustrates the superb integration capability of
393 libraries for client-side scripting in web browsers. The modularity and simplicity of client-side
394 web programming allows easy adaptation to other kinds of experimental data; for example to
395 residue-specific paramagnetic relaxation enhancements as done by Prof. Rasia (IBR-CONICET-
396 UNR, Argentina) at https://rrasia.altervista.org/HYL1_1-2/Hyl1_12_minima.html.

397   Another example, presented in Figure 4B shows how AR can help to explore residue-
398 residue contact predictions from residue coevolution data. Such predictions provide useful
399 restraints in modern approaches for modeling proteins and their complexes (Simkovic et al., 2017;
400 Abriata et al., 2018; Abriata, Tamò & Dal Peraro, 2019), but often include a fraction of false
401 positives that introduce incorrect information if undetected. Interactive inspection of residue-
402 residue contact predictions could help to actively detect false positives through human intervention
403 before the actual restraint-guided modeling proceeds. The example in Figure 4B shows contacts
404 predicted from coevolution analysis of large sequence alignments for the pair of proteins in chains
405 A and B of PDB ID 1QOP, taken from the Gremlin server (Ovchinnikov, Kamisetty & Baker,
406 2014). Each protein is driven by one marker, and the predicted contacts are overlaid as dotted lines
407 connecting the intervening pairs of residues. These lines are colored green, olive and red according
408 to decreasing coevolution score as in the Gremlin website, and their widths reflect in real time the

409   distance between pairs of residues, presumably minimal when contacts are satisfied if the
410   prediction is true.

411   The last prototype application shows rudimentary handling of highly disordered protein
412   regions, in this case to test how a flexible linker made of 6 glycine residues restricts the space
413   available for exploration and possible docking poses of two well-folded domains (Figure 5A-C).
414   Each folded domain (ubiquitin and ubiquitin-interacting motif, taken from PDB ID 2D3G) is
415   modeled at a resolution of 2 spheres per residue, one centered at the backbone's alpha carbon and
416   one at the center of mass of the sidechain (*i.e.* a description slightly coarser than that of the
417   MARTINI force field (Marrink et al., 2007)). All spheres representing residues of the folded
418   domains are kept in fixed positions relative to their AR marker, and have radii assigned as the
419   cubic root of the amino acid volumes to roughly respect the relative amino acid sizes (taken from
420   (Abriata, Palzkill & Dal Peraro, 2015)). The glycine residues of the flexible linker are modeled as
421   single spheres centered at the alpha carbons with their radii set to the cubic root of glycine's
422   volume. Using Cannon.js, the spheres representing the glycine residues of the flexible linker (in
423   orange in Figure 5) are allowed to move freely but under a fixed-distance constraint from each
424   other and from the corresponding ends of the folded domains. This very simple model can help to
425   answer questions related to the separation of the anchor points and the maximal extension of the
426   linker when straight: How far apart can the two proteins go with the given linker containing 6
427   residues? Can both proteins be docked through certain interfaces keeping the linker in a relaxed
428   configuration? The user's investigations are assisted by on-the-fly estimation of entropy associated
429   to given extensions of the linkers, estimated with a worm-like chain model from polymer physics
430   (Marko & Siggia, 1995), and by an estimation of the strain experienced by the linker when the
431   user pulls its glycine residues apart beyond their equilibrium distance.

432

433   ## Discussion
434   Achieving seamless integration of immersive visualizations, haptic interfaces and chemical
435   computation stands as one of the key "grand challenges" for the simulation of matter in the 21st
436   century (Aspuru-Guzik, Lindh & Reiher, 2018). Such integration is expected to help us to more
437   easily grasp and explore molecular properties and to efficiently navigate chemical information. In
438   the last two decades several works introduced different ways to achieve AR and VR, as presented
439   in the Introduction section. In education, such tools could replace or complement physical (such

440   as plastic-made) modeling kits, augmenting them with additional information about forces,

441   charges, electron distributions, data facts, etc. In research, such tools could help better visualize

442   and probe molecular structure, simulate expected outcomes of experiments and test models and

443   simulated data against experimental data, etc., all through intuitive cues and fluent human-

444   computer interactions.

445   This article introduced a minimal set of building blocks and code for developing marker-

446   based AR applications for molecular modeling in web browsers using regular computers. Since

447   web AR is a new and emerging technology, it is reasonable to identify some limitations. Naturally,

448   such web apps cannot currently match the graphics quality and interfacing capabilities of programs

449   prepared for specialized AR/VR hardware, and the use of a webcam to follow markers can result

450   in unstable tracking under certain conditions compared to setups that use *ad hoc* devices. Another

451   limitation is that support for AR in smartphones is not uniform, and varies greatly between devices.

452   Currently, only laptops and desktop computers provide a consistent experience. Lastly, content

453   developers must routinely verify that the web apps continue to work in major browsers after

454   updates, and also validate them against new W3C standards, APIs and libraries. On the bright side,

455   these web apps have some important advantages. First, all examples presented here rely only on

456   client-side programming; therefore, applications need only be uploaded to regular webserver to

457   become available to the world. Second, they do not require client-side plugins, so are supported

458   "out of the box" by standard web browsers. Third, the modularity of JavaScript and the availability

459   of several ready-to-use libraries greatly simplify development of new content. Fourth, users of

460   these applications do not need to worry about updates, as they always receive the latest version

461   when they reload the page. All in all, these positive features will favor adoption of browser based

462   AR/VR for educational purposes over alternatives that require specialized software and hardware.

463

Future Perspectives

465   For educational applications, the next stage would be to develop actual content of value for

466   teachers and students. The simplest content could merely provide visualizations to explore

467   molecular geometries, surfaces, orbitals, etc., with specific sets of demos to assist learning of key

468   concepts such as chirality, organic molecules, metal coordination complexes and

469   biomacromolecular structures, to mention just a few cases. By adding mechanics, more complex

470   demos could be created where students could for example interactively explore torsion angles as

471 in the textbook cases of probing the energy landscape of rotation around the central C-C bond of

472 butane, or swapping between chair and boat conformations of 6-member rings, exploring hydrogen

473 bonding patterns between neighbor beta strands in a protein, etc. Importantly, every single student

474 having a computer at hand could use these apps, not only at the school/university but also at home,

475 therefore this could become an actual learning tool of full-time, individual use. The possibility of

476 reaching the masses with this kind of web technologies for AR-based molecular modeling in turn

477 opens up the opportunity of performing large-scale evaluations of their actual impact in education.

478   As actively investigated by others, there is also a need to explore if full working programs

479 for AR-based molecular modeling may actually become powerful enough to also assist research.

480 Here again, web-based tools like those discussed in this article could help to carry out such tests

481 at large scales. Some of the prototypes presented here advance possible uses in research, as in the

482 simulation of data from protein-protein docking poses and comparison to the corresponding

483 experimental data in real time. However, some issues should be addressed before creating fully-

484 fledged web programs for research: (i) improving AR marker detection and tracking to stabilize

485 inputs (Q. H. Gao et al., 2017), (ii) developing some kind of AR marker that is clickable so that

486 users can drag and drop objects in space (hitherto unexplored), (iii) improving graphics, where the

487 possibility of adapting existing web molecular graphics like NGL (Rose & Hildebrand, 2015),

488 3dmol (Rego & Koes, 2015), Litemol (Sehnal et al., 2017), Mol*(Sehnal et al., 2018), JSmol

489 (Hanson et al., 2013), etc. is particularly enticing, and (iv) developing force fields that correctly

490 handle molecular dynamics and energetics for different tasks, which may imply different levels of

491 granularity for different applications. Another global improvement, also important for pedagogical

492 applications, would be incorporating proper object occlusion, which is still non-trivial and subject

493 of studies in the AR community (M. M. Shah, H. Arshad & R. Sulaiman, 2012; J. Gimeno et al.,

494 2018).

495   Some further directions that could be explored in the near future are fluently connecting

496 through an AR experience users in physically distant locations, so that they can collaborate on a

497 research project or teach/learn at a distance. Adapting future standards for commodity AR/VR in

498 smartphones (plugged into cardboard goggles for stereoscopy) is also worth exploring as this

499 would lead to an even more immersive experience than with the mirror-like apps proposed here.

500 However, since smartphones are more limited in power, their use for AR molecular modeling may

501 require coupling to external computer power. Last, pushing the limits towards fully immersive

502   visual analytics for molecular modeling, and especially thinking about solutions useful for

503   research, a few especially enticing additions include support for haptic devices for force feedback,

504   AR without markers (i.e. just by tracking the users' hands and fingers) and considering occlusion,

505   and as described above, the capability to respond to visual or audio cues by automatically checking

506   databases and mining literature to propose relevant pieces of information.

507

508   **Acknowledgements**

509   I acknowledge all the members of the communities that develop the client-side web programming

510   tools used here, as well as the very helpful communities who contribute to their improvement, bug

511   detection and correction, documentation, and online help. Special thanks to J. Ethienne (AR.js

512   developer), Nicolò Carpignoli (AR.js maintainer), D. McCurdy (Google), T. Ater (Annyang

513   developer), L. Stemkoski (Adelphi U., USA), A. Herráez (U. Alcalá, Spain) and A. Papoutsaki

514   (Pomona College, USA) for assistance. I also acknowledge numerous researchers from the

515   physics, chemistry and biology communities who have provided ideas, suggestions, examples and

516   support, especially L. Krapp, S. Träger, M. Dal Peraro, F.G. van der Goot and M.E. Zaballa (EPFL,

517   Switzerland), A. Barducci (CBS, France), D. Moreno and G. Frattini (IQUIR-CONICET-UNR,

518   Argentina), and R. Rasia (IBR-CONICET-UNR, Argentina). Finally, I greatly acknowledge the

519   extremely helpful revisions from the Editor and two reviewers.

520

521

522   **References**

523

524   Abriata LA. 2017a. Web Apps Come of Age for Molecular Sciences. *Informatics* 4:28. DOI:

525   10.3390/informatics4030028.

526   Abriata LA. 2017b. Concurrent interactive visualization and handling of molecular structures

527   over the Internet in web browsers. *ArXiv e-prints*.

528   Abriata LA, Palzkill T, Dal Peraro M. 2015. How structural and physicochemical determinants

529   shape sequence constraints in a functional enzyme. *PloS One* 10:e0118684. DOI:

530   10.1371/journal.pone.0118684.

531   Abriata LA, Rodrigues JPGLM, Salathé M, Patiny L. 2017. Augmenting research, education and

532   outreach with client-side web programming. *Trends in Biotechnology*.

533   Abriata LA, Tamò GE, Dal Peraro M. 2019. A further leap of improvement in tertiary structure

534   prediction in CASP13 prompts new routes for future assessments. *Proteins*. DOI:

535   10.1002/prot.25787.

536   Abriata LA, Tamo G, Monastyrskyy M, Kryshtafovych A, Dal Peraro M. 2018. Assessment of

537   hard target modeling in CASP12 reveals an emerging role of alignment-based contact

538   prediction methods. *Proteins: Structure, Function, and Bioinformatics* 86 Suppl 1:97–

539   112.

540   Aspuru-Guzik A, Lindh R, Reiher M. 2018. The Matter Simulation (R)evolution. *ACS Central*

541   *Science* 4:144–152. DOI: 10.1021/acscentsci.7b00550.

542   Ater T. 2019. *Annyang*. https://github.com/TalAter/annyang.

543   Bach B, Sicat R, Beyer J, Cordeil M, Pfister H. 2018. The Hologram in My Hand: How Effective

544   is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented

545   Reality? *IEEE transactions on visualization and computer graphics* 24:457–467. DOI:

546   10.1109/TVCG.2017.2745941.

547   Balo AR, Wang M, Ernst OP. 2017. Accessible virtual reality of biomolecular structural models

548   using the Autodesk Molecule Viewer. *Nature Methods* 14:1122–1123. DOI:

549   10.1038/nmeth.4506.

550   Bartók AP, De S, Poelking C, Bernstein N, Kermode JR, Csányi G, Ceriotti M. 2017. Machine

551   learning unifies the modeling of materials and molecules. *Science Advances* 3. DOI:

552   10.1126/sciadv.1701816.

553   Berry C, Board J. 2014. A Protein in the palm of your hand through augmented reality.

554   *Biochemistry and Molecular Biology Education: A Bimonthly Publication of the*

555   *International Union of Biochemistry and Molecular Biology* 42:446–449. DOI:

556   10.1002/bmb.20805.

557 Bertini I, Turano P, Vila AJ. 1993. NMR of paramagnetic metalloproteins. *Chem.Rev.* 93:2833–

558   2932.

559 Bolopion A, Cagneau B, Redon S, Régnier S. 2009. *Haptic feedback for molecular simulation*.

560   DOI: 10.1109/IROS.2009.5354256.

561 Bolopion A, Cagneau B, Redon S, Régnier S. 2010. Comparing position and force control for

562   interactive molecular simulators with haptic feedback. *Journal of Molecular Graphics &*

563   *Modelling* 29:280–289. DOI: 10.1016/j.jmgm.2010.06.003.

564 Debye P. 1915. Zerstreuung von Röntgenstrahlen. *Annalen der Physik* 351:809–823. DOI:

565   10.1002/andp.19153510606.

566 Gillet A, Sanner M, Stoffler D, Goodsell D, Olson A. 2004. Augmented reality with tangible auto-

567   fabricated models for molecular biology applications.

568 Gillet A, Sanner M, Stoffler D, Olson A. 2005. Tangible interfaces for structural molecular

569   biology. *Structure (London, England: 1993)* 13:483–491. DOI: 10.1016/j.str.2005.01.009.

570 Goddard TD, Brilliant AA, Skillman TL, Vergenz S, Tyrwhitt-Drake J, Meng EC, Ferrin TE.

571   2018a. Molecular Visualization on the Holodeck. *Journal of Molecular Biology*. DOI:

572   10.1016/j.jmb.2018.06.040.

573 Goddard TD, Huang CC, Meng EC, Pettersen EF, Couch GS, Morris JH, Ferrin TE. 2018b.

574   UCSF ChimeraX: Meeting modern challenges in visualization and analysis. *Protein*

575   *Science: A Publication of the Protein Society* 27:14–25. DOI: 10.1002/pro.3235.

576 Hanson RM, Prilusky J, Renjian Z, Nakane T, Sussman JL. 2013. JSmol and the Next-

577   Generation Web-Based Representation of 3D Molecular Structure as Applied to

578   Proteopedia. *Israel Journal of Chemistry* 53:207–216. DOI: 10.1002/ijch.201300024.

579 Hendler J. 2003. Science and the Semantic Web. *Science* 299:520. DOI:

580   10.1126/science.1078874.

581 Hirano S, Kawasaki M, Ura H, Kato R, Raiborg C, Stenmark H, Wakatsuki S. 2006. Double-

582   sided ubiquitin binding of Hrs-UIM in endosomal protein sorting. *Nature Structural &*

583   *Molecular Biology* 13:272–277. DOI: 10.1038/nsmb1051.

584 Hirst JD, Glowacki DR, Baaden M. 2014. Molecular simulations and visualization: introduction

585   and overview. *Faraday Discuss.* 169:9–22. DOI: 10.1039/C4FD90024C.

586 Humphrey W, Dalke A, Schulten K. 1996. VMD – Visual Molecular Dynamics. *Journal of*

587   *Molecular Graphics* 14:33–38.

588 J. Gimeno, S. Casas, C. Portalés, M. Fernádez. 2018. Addressing the Occlusion Problem in

589   Augmented Reality Environments with Phantom Hollow Objects. In: *2018 IEEE*

590   *International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*.

591   21–24. DOI: 10.1109/ISMAR-Adjunct.2018.00024.

592 Jiang C, Jin X. 2017. Quick Way to Port Existing C/C++ Chemoinformatics Toolkits to the Web

593   Using Emscripten. *Journal of Chemical Information and Modeling*. DOI:

594   10.1021/acs.jcim.7b00434.

595 Kato H. 1999. *ARToolKit*. http://www.hitl.washington.edu/artoolkit/.

596 Lee J, Kim J-I, Kang L-W. 2012. A Collaborative Molecular Modeling Environment Using a

597   Virtual Tunneling Service. *Journal of Biomedicine and Biotechnology* 2012:7.

598 M. DiPierro. 2018. The Rise of JavaScript. *Computing in Science & Engineering* 20:9–10. DOI:

599   10.1109/MCSE.2018.011111120.

600 M. Fjeld, B. M. Voegtli. 2002. Augmented Chemistry: an interactive educational workbench. In:

601   *Proceedings. International Symposium on Mixed and Augmented Reality*. 259–321. DOI:

602   10.1109/ISMAR.2002.1115100.

603 M. Krichenbauer, G. Yamamoto, T. Taketom, C. Sandor, H. Kato. 2018. Augmented Reality

604   versus Virtual Reality for 3D Object Manipulation. *IEEE Transactions on Visualization*

605   *and Computer Graphics* 24:1038–1048. DOI: 10.1109/TVCG.2017.2658570.

606   M. M. Shah, H. Arshad, R. Sulaiman. 2012. Occlusion in augmented reality. In: *2012 8th*
607   *International Conference on Information Science and Digital Content Technology*
608   *(ICIDT2012)*. 372–378.

609   Maier P, Klinker G. 2013. Augmented Chemical Reactions: 3D Interaction Methods for
610   Chemistry. *International Journal of Online Engineering* 9.

611   Maier P, Tönnis M, GudrunKlinker D. 2009. Dynamics in Tangible Chemical Reactions. In:
612   *World Academy of Science, Engineering and Technology*.

613   Marko JF, Siggia ED. 1995. Statistical mechanics of supercoiled DNA. *Phys. Rev. E* 52:2912–
614   2938. DOI: 10.1103/PhysRevE.52.2912.

615   Marrink SJ, Risselada HJ, Yefimov S, Tieleman DP, de Vries AH. 2007. The MARTINI force
616   field: coarse grained model for biomolecular simulations. *The Journal of Physical*
617   *Chemistry. B* 111:7812–7824. DOI: 10.1021/jp071097f.

618   Martínez-Hung H, García-López CA, Escalona-Arranz JC. 2016. Augmented Reality Models
619   Applied to the Chemistry Education on the University (article in Spanish). *Revista*
620   *Cubana de Química* 29:13–25.

621   Matthews D. 2018. Virtual-reality applications give science a new dimension. *Nature* 557:127–
622   128. DOI: 10.1038/d41586-018-04997-2.

623   Matthews N, Kitao A, Laycock S, Hayward S. 2019. Haptic-Assisted Interactive Molecular
624   Docking Incorporating Receptor Flexibility. *Journal of Chemical Information and*
625   *Modeling* 59:2900–2912. DOI: 10.1021/acs.jcim.9b00112.

626   Moore EB, Chamberlain JM, Parson R, Perkins KK. 2014. PhET Interactive Simulations:
627   Transformative Tools for Teaching Chemistry. *Journal of Chemical Education* 91:1191–
628   1197. DOI: 10.1021/ed4005084.

629   Müller C, Krone M, Huber M, Biener V, Herr D, Koch S, Reina G, Weiskopf D, Ertl T. 2018.
630   Interactive Molecular Graphics for Augmented Reality Using HoloLens. *Journal of*
631   *Integrative Bioinformatics* 15. DOI: 10.1515/jib-2018-0005.

632   O'Connor M, Deeks HM, Dawn E, Metatla O, Roudaut A, Sutton M, Thomas LM, Glowacki BR,

633   Sage R, Tew P, Wonnacott M, Bates P, Mulholland AJ, Glowacki DR. 2018. Sampling

634   molecular conformations and dynamics in a multiuser virtual reality framework. *Science*

635   *Advances* 4:eaat2731. DOI: 10.1126/sciadv.aat2731.

636   O'Donoghue SI, Gavin A-C, Gehlenborg N, Goodsell DS, Hériché J-K, Nielsen CB, North C,

637   Olson AJ, Procter JB, Shattuck DW, Walter T, Wong B. 2010. Visualizing biological data-

638   now and in the future. *Nature Methods* 7:S2-4. DOI: 10.1038/nmeth.f.301.

639   Ovchinnikov S, Kamisetty H, Baker D. 2014. Robust and accurate prediction of residue-residue

640   interactions across protein interfaces using evolutionary information. *eLife* 3:e02030.

641   DOI: 10.7554/eLife.02030.

642   Pafilis E, O'Donoghue SI, Jensen LJ, Horn H, Kuhn M, Brown NP, Schneider R. 2009. Reflect:

643   augmented browsing for the life scientist. *Nature Biotechnology* 27:508–510. DOI:

644   10.1038/nbt0609-508.

645   Papoutsaki A, Sangkloy P, Laskey J, Daskalova N, Huang J, Hays J. 2016. WebGazer:

646   Scalable Webcam Eye Tracking Using User Interactions. In: *Proceedings of the 25th*

647   *International Joint Conference on Artificial Intelligence*. 3839–3845.

648   Paruzzo FM, Hofstetter A, Musil F, De S, Ceriotti M, Emsley L. Chemical Shifts in Molecular

649   Solids by Machine Learning. *ArXiv* 1805:11541.

650   Pence HE, Williams AJ, Belford RE. 2015. New Tools and Challenges for Chemical Education:

651   Mobile Learning, Augmented Reality, and Distributed Cognition in the Dawn of the Social

652   and Semantic Web. In: *Chemistry Education*. Wiley-VCH Verlag GmbH & Co. KGaA,

653   693–734. DOI: 10.1002/9783527679300.ch28.

654   Q. H. Gao, T. R. Wan, W. Tang, L. Chen. 2017. A Stable and Accurate Marker-Less Augmented

655   Reality Registration Method. In: *2017 International Conference on Cyberworlds (CW)*.

656   41–47. DOI: 10.1109/CW.2017.44.

657   Ratamero EM, Bellini D, Dowson CG, Römer RA. 2018. Touching proteins with virtual bare

658   hands : Visualizing protein-drug complexes and their dynamics in self-made virtual

659   reality using gaming hardware. *Journal of Computer-Aided Molecular Design* 32:703–

660   709. DOI: 10.1007/s10822-018-0123-0.

661   Rebholz-Schuhmann D, Oellrich A, Hoehndorf R. 2012. Text-mining solutions for biomedical

662   research: enabling integrative biology. *Nature Reviews. Genetics* 13:829–839. DOI:

663   10.1038/nrg3337.

664   Rego N, Koes D. 2015. 3Dmol.js: molecular visualization with WebGL. *Bioinformatics (Oxford,*

665   *England)* 31:1322–1324. DOI: 10.1093/bioinformatics/btu829.

666   Rose AS, Hildebrand PW. 2015. NGL Viewer: a web application for molecular visualization.

667   *Nucleic Acids Research* 43:W576-579. DOI: 10.1093/nar/gkv402.

668   Russel D, Lasker K, Webb B, Velázquez-Muriel J, Tjioe E, Schneidman-Duhovny D, Peterson

669   B, Sali A. 2012. Putting the pieces together: integrative modeling platform software for

670   structure determination of macromolecular assemblies. *PLoS biology* 10:e1001244. DOI:

671   10.1371/journal.pbio.1001244.

672   S. Doutreligne, T. Cragnolini, S. Pasquali, P. Derreumaux, M. Baaden. 2014. UnityMol:

673   Interactive scientific visualization for integrative biology. In: *2014 IEEE 4th Symposium*

674   *on Large Data Analysis and Visualization (LDAV)*. 109–110. DOI:

675   10.1109/LDAV.2014.7013213.

676   Sehnal D, Deshpande M, Svobodová Vařeková R, Mir S, Berka K, Midlik A, Pravda L, Velankar

677   S, Koča J. 2017. LiteMol suite: interactive web-based visualization of large-scale

678   macromolecular structure data. *Nature Methods* in press.

679   Sehnal D, Rose AS, Koča J, Burley SK, Velankar S. 2018. Mol*: Towards a Common Library

680   and Tools for Web Molecular Graphics. In: *Proceedings of the Workshop on Molecular*

681   *Graphics and Visual Analysis of Molecular Data*. MolVA '18. Goslar Germany, Germany:

682   Eurographics Association, 29–33.

683   Sekhavat YA, Zarei H. 2016. Enhancing the Sense of Immersion and Quality of Experience in

684   Mobile Games Using Augmented Reality. *Journal of Computing and Security* 3:53–62.

685   Simkovic F, Ovchinnikov S, Baker D, Rigden DJ. 2017. Applications of contact predictions to

686   structural biology. *IUCrJ* 4:291–300. DOI: 10.1107/S2052252517005115.

687   Smith JS, Isayev O, Roitberg AE. 2017. ANI-1: an extensible neural network potential with DFT

688   accuracy at force field computational cost. *Chemical Science* 8:3192–3203. DOI:

689   10.1039/c6sc05720a.

690   Sommer B, Baaden M, Krone M, Woods A. 2018. From Virtual Reality to Immersive Analytics in

691   Bioinformatics. *Journal of Integrative Bioinformatics* 15. DOI: 10.1515/jib-2018-0043.

692   Stocks MB, Hayward S, Laycock SD. 2009. Interacting with the biomolecular solvent accessible

693   surface via a haptic feedback device. *BMC structural biology* 9:69. DOI: 10.1186/1472-

694   6807-9-69.

695   Stocks MB, Laycock SD, Hayward S. 2011. Applying forces to elastic network models of large

696   biomolecules using a haptic feedback device. *Journal of Computer-Aided Molecular*

697   *Design* 25:203–211. DOI: 10.1007/s10822-010-9410-0.

698   Stone J. 2019. *VMD support for VR and Interactive MD*.

699   https://www.ks.uiuc.edu/Research/vmd/allversions/interactive_MD.html:

700   https://www.ks.uiuc.edu/Research/vmd/allversions/interactive_MD.html.

701   Stovgaard K, Andreetta C, Ferkinghoff-Borg J, Hamelryck T. 2010. Calculation of accurate small

702   angle X-ray scattering curves from coarse-grained protein models. *BMC bioinformatics*

703   11:429. DOI: 10.1186/1471-2105-11-429.

704   V. Pimentel, B. G. Nickerson. 2012. Communicating and Displaying Real-Time Data with

705   WebSocket. *IEEE Internet Computing* 16:45–53. DOI: 10.1109/MIC.2012.64.

706   Vega Garzón JC, Magrini ML, Galembeck E. 2017. Using augmented reality to teach and learn

707   biochemistry. *Biochemistry and Molecular Biology Education: A Bimonthly Publication of*

708   *the International Union of Biochemistry and Molecular Biology*. DOI:

709   10.1002/bmb.21063.

710   Wiebrands M, Malajczuk CJ, Woods AJ, Rohl AL, Mancera RL. 2018. Molecular Dynamics

711   Visualization (MDV): Stereoscopic 3D Display of Biomolecular Structure and Interactions

712   Using the Unity Game Engine. *Journal of Integrative Bioinformatics* 15. DOI: 10.1515/jib-

713   2018-0010.

714   Wollacott AM, Merz KM. 2007. Haptic applications for molecular structure manipulation. *Journal*

715   *of Molecular Graphics & Modelling* 25:801–805. DOI: 10.1016/j.jmgm.2006.07.005.

716   Wolle P, Müller MP, Rauh D. 2018. Augmented Reality in Scientific Publications—Taking the

717   Visualization of 3D Structures to the Next Level. *ACS Chemical Biology* 13:496–499.

718   DOI: 10.1021/acschembio.8b00153.

719   Xu K, Liu N, Xu J, Guo C, Zhao L, Wang H-W, Zhang QC. 2019. VRmol: an Integrative Cloud-

720   Based Virtual Reality System to Explore Macromolecular Structure. *bioRxiv*:589366.

721   DOI: 10.1101/589366.

722   Y. Gaffary, B. Le Gouis, M. Marchal, F. Argelaguet, B. Arnaldi, A. Lécuyer. 2017. AR Feels

723   "Softer" than VR: Haptic Perception of Stiffness in Augmented versus Virtual Reality.

724   *IEEE Transactions on Visualization and Computer Graphics* 23:2372–2377. DOI:

725   10.1109/TVCG.2017.2735078.

726   Yang S, Mei B, Yue X. 2018. Mobile Augmented Reality Assisted Chemical Education: Insights

727   from Elements 4D. *Journal of Chemical Education* 95:1060–1062. DOI:

728   10.1021/acs.jchemed.8b00017.

729

Figure 1

Commodity augmented reality in web browsers

(A) AR markers "Hiro" and "Kanji" are built into AR.js and its A-Frame wrap. Figure S1 shows them ready to print at various sizes. Custom markers including cube markers are also implementable. (B) In the proposed web apps the AR library feeds coordinate information to the graphics libraries and other algorithms. (C) The web pages must be hosted in servers compatible with the https protocol, but their content then runs exclusively on the computer.

Figure 2

Different implementations of WebGL for AR in web browsers.

(A) 2-bromo-butane enantiomers R and S displayed on Hiro and Kanji markers, respectively, from Wavefront objects produced in VMD. (B) A protein complex rendered as cartoons with small molecule ligands shown as sticks on the Hiro marker (PDB ID 1VYQ). (C) A double-stranded segment of DNA (sticks) bound to a homeodomain protein (cartoon) spinning on the Kanji marker (PDB ID 1FJL). Both B and C were rendered from VMD Wavefront objects. (D) Volumetric map of a bacteriophage attached to a portion of the host's cell wall as determined through electron microscopy (EMDB ID 9010) and prepared as VMD Wavefront object showing the capsid in grey, the needle in blue and the membranes in orange. (E) A small peptide inside its electron density map as determined by x-ray diffraction (PDB ID 3HYD). (F) HOMO and LUMO orbitals from NH_3 and BH_3 molecules, rendered from Wavefront objects. (G) Representation of an amphipathic alpha helix built from primitives, viewable on the Kanji marker. (H) Use of a cube marker (made up of 6 different AR markers in its faces) to load any molecule in PDB format and handle and visualize it in 3D. Graphics built from Three.js primitives. The example also uses Cannon.js to simulate rigid body dynamics by fixing the distances between atoms separated by 1 or 2 bonds but allowing rotations, in analogy to plastic-made molecular modeling kits.

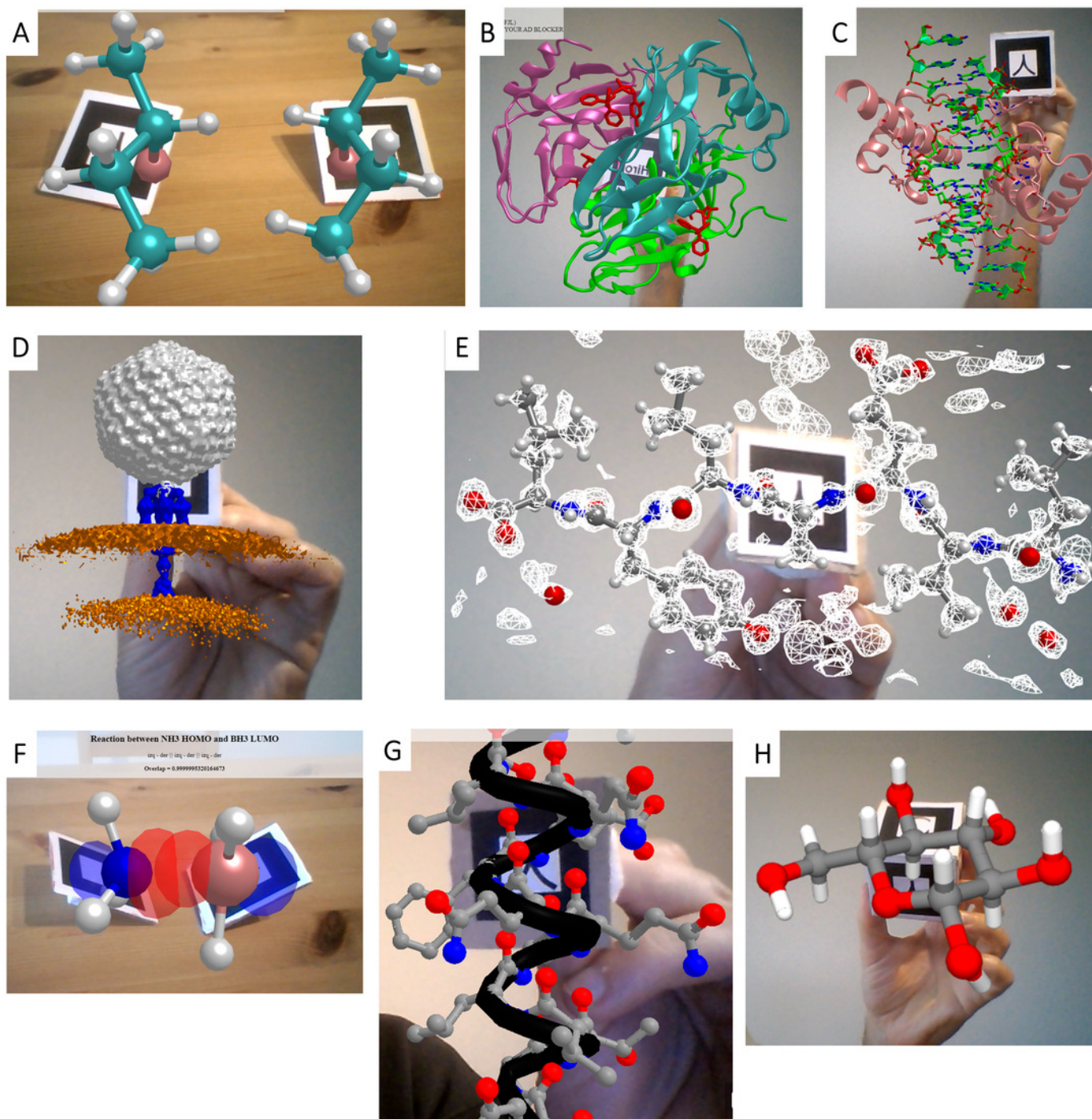



Figure 3

Introducing interactivity

(A) A lysine and a glutamate side chains attached to different AR markers, whose coordinates are processed in real time to deliver distance and electrostatic potential between charged groups and to calculate and display clashes. Graphics achieved with A-Frame polyfills. In this example the ratio of protonated lysine to glutamate exchanging dynamically is set to 70/30 = 2.33, i.e. favoring protonated lysine as the actual water chemistry dictates but shifted orders of magnitude from the real ratio of acidic constants so that the user can observe temporal protonation of both amino acids in the timescale of work. (B) A Diels-Alder reaction, taking place once the user has moved the reagents close enough in space; this example further helps to visualize fused cycles (as the diene reagent was a cyclic molecule itself). (C) Example of interactively probing experimental observables: as a probe atom (black sphere) is moved around a paramagnetic center, the user sees the paramagnetic effects simulated at the location of the probe atom in real time.

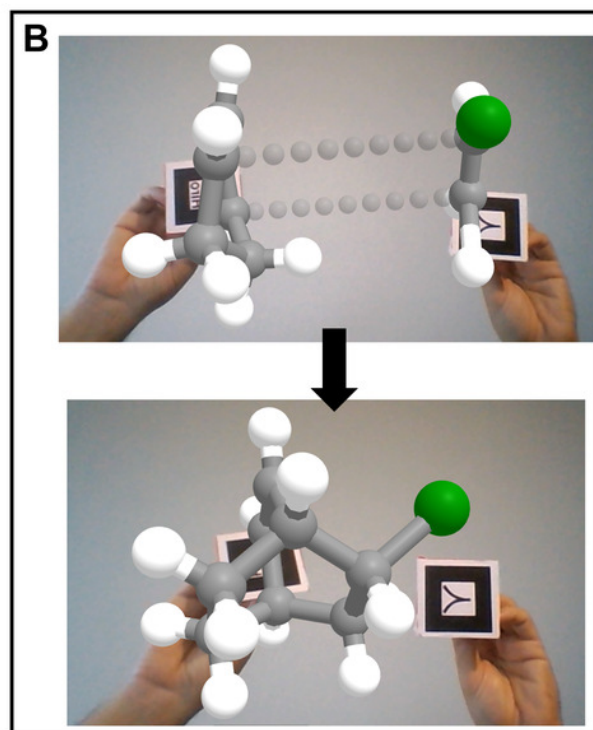
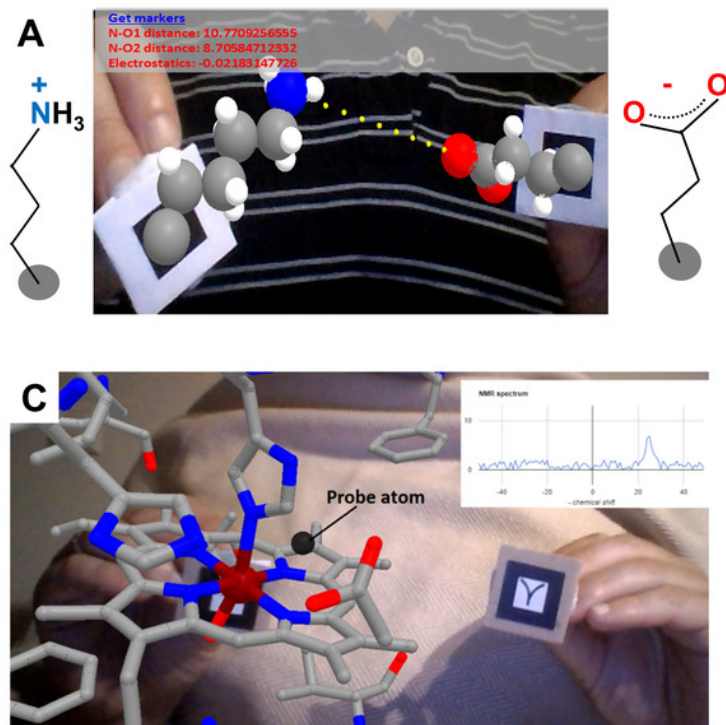



Figure 4

Interactivity in biological macromolecules

(A) Ubiquitin and ubiquitin-interacting motif (red and blue respectively, taken from PDB ID 2D3G) driven in 3D with two AR markers, as the web app computes the predicted SAXS profile and displays it overlaid on top of a purported experimental profile together with a metric of the fit quality. (B) Interactive exploration of contacts predicted between two proteins (here from coevolution data) before and after manual docking. This example is 1QOP from Ovchinnikov et al, where contacts predicted with high score are colored green, contacts of intermediate confidence are olive, and the first contact of low probability is shown red (as taken from the original data). The thickness of the contact lines indicates distance, such that thin lines indicate the residues are close in space. Notice how the red contact, which has low probability, remains thicker than the well-scored contacts (green) upon manual docking.

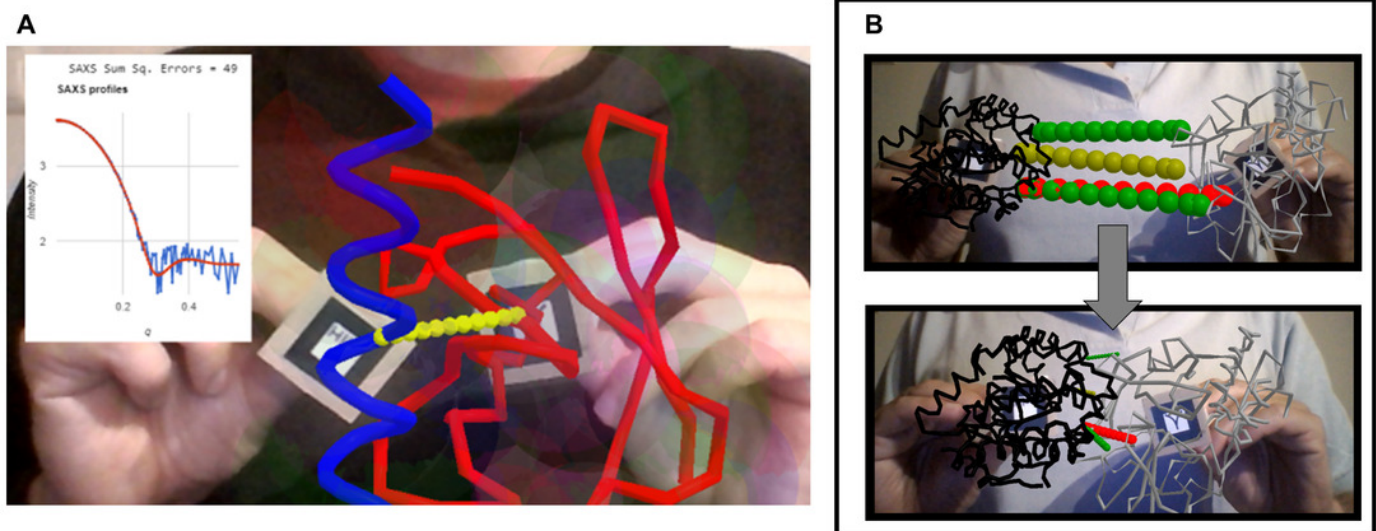
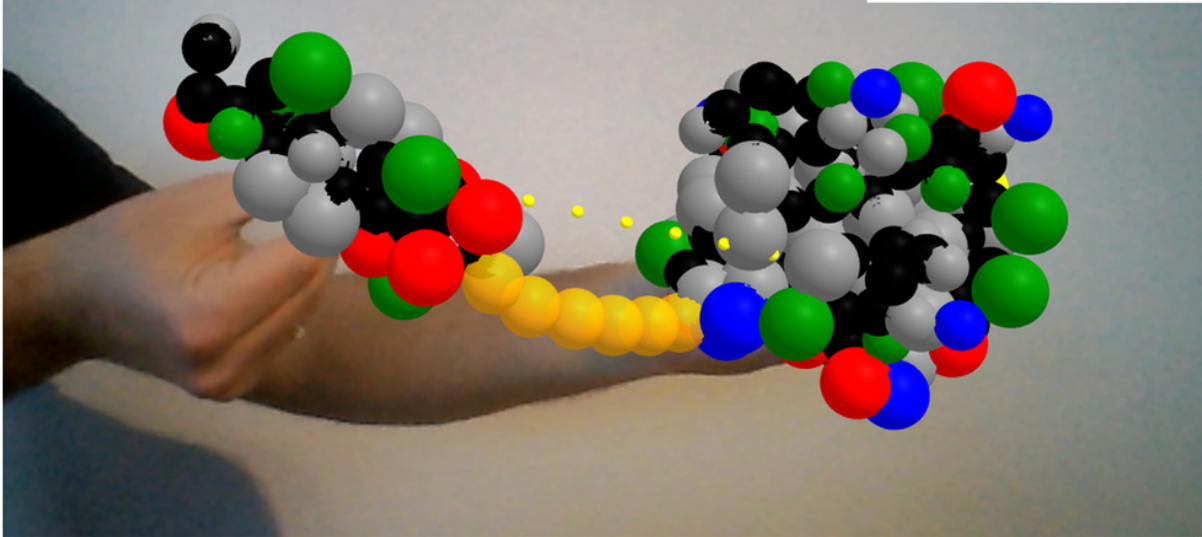
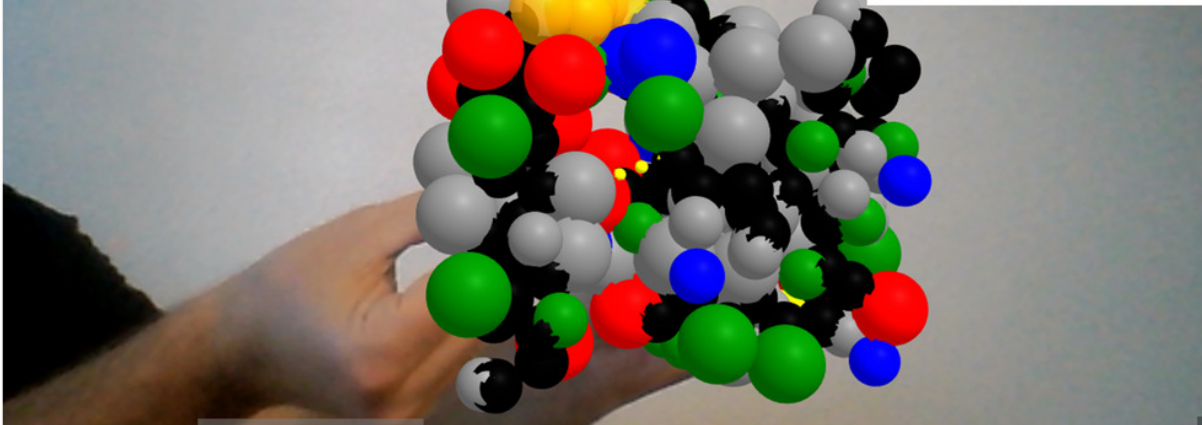
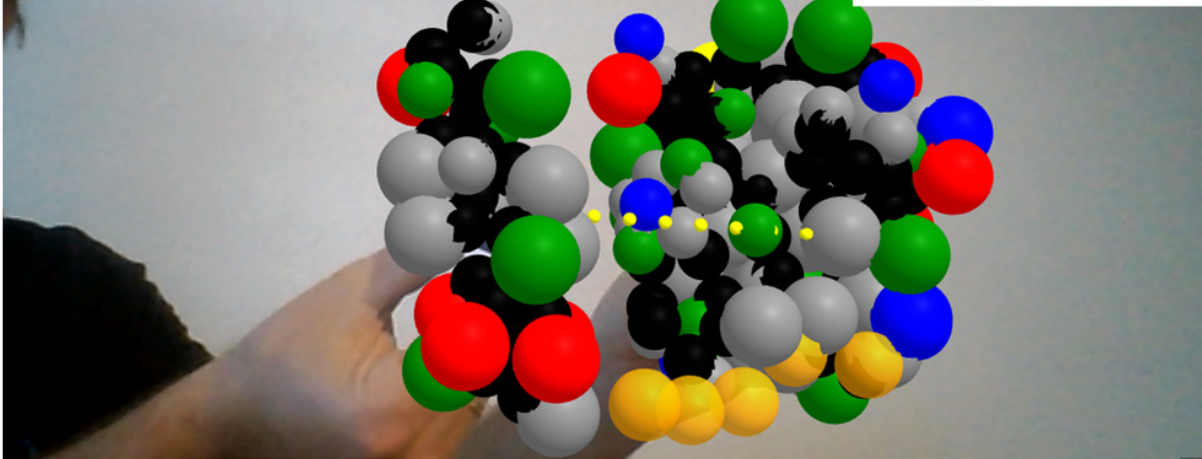



Figure 5

Dynamics of highly disordered protein linkers modeled with rigid-body mechanics

Ubiquitin and ubiquitin-interacting motif (PDB ID 2D3G) modelled as 2-4 beads per residue, colored by physicochemical properties (grey=hydrophobic, red=negative, blue=positive, green=polar uncharged; backbone beads in black). The domains are driven independently in 3D with two AR markers. They are connected through a flexible linker of six backbone-sized beads (orange) whose dynamics are treated with the Cannon.js rigid-body force field. The web app reports in real time the distance between the centers of both domains, the entropy of the linker based on a worm-like chain model, and the linker strain computed from deviations of distances between consecutive linker beads from an equilibrium distance. (A) The two domains extended as much as possible while keeping the linker relaxed (although entropically unfavoured) illustrates the maximum possible separation with the given linker is of around 40 Å. (B) The binding pose between the 2 domains is geometrically feasible as it keeps the linker relaxed. (C) This other binding pose is unachievable with a linker of this length.

A   Distance = 40.2 Å   Linker strain = 7 Å²
Entropy = 13

B   Distance = 17.7 Å   Linker strain = 2 Å²
Entropy = 10

C   Distance = 26.2 Å   Linker strain = 52 Å²
Entropy = 2

Links to all web examples arranged by figure

See https://lucianoabriata.altervista.org/papersdata/tablepeerjcs2019.html for an online version of the table which also includes links to several video demonstrations.

1   **Table 1. Links to all web examples arranged by figure.** See
2   https://lucianoabriata.altervista.org/papersdata/tablepeerjcs2019.html for an online version of the table
3   which also includes links to several video demonstrations.

Figure 2	
2A	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/2brbutane.html
2B,C	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/pdb-1vyq-1fjl.html
2D	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/bacteriophage.html
2E	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/xray3hyd.html
2F	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/bh3nh3orb.html
2G	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/ubiquitinandNESatomistic.html
2H	https://lucianoabriata.altervista.org/jsinscience/arjs/jsartoolkit5/pdbloader6.html

Figure 3	
3A	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/smallmolclashdetection.html
	and https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/smallmolprotontransfer.html
3B	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/smallmoldielsalder.html
3C	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/metmyoglobinfe3pcshift.html

Figure 4	
4A	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/ubiquitinuimffvoicesaxs.html
4B-C	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/coevol_1qop.html

Figure 5	
5A-C	https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/ubiquitin-uim-cannon.html

Requirements and troubleshooting for developers and users

1   **Table 2. Requirements and troubleshooting for developers and users.**

Developers

Software and hardware requirements

* Ensure using https URLs; otherwise webcams will not activate.
* Free web hosting services work, as web pages only need to be hosted but run in the client.
* Given the regular updates in w3c standards, APIs and libraries, routine tests are recommended to ensure normal functioning.
* Examples from this paper were verified to work "out of the box" on Safari in multiple MacOS 10.x versions and on multiple Chrome and Firefox versions in Windows 8, Windows 10, Linux RedHat Enterprise Edition, Ubuntu and ArchLinux.
* Currently limited and heterogeneous support in tablets and smartphones, these devices are not recommended.

Users

Software and hardware requirements

* Need a webcam- and internet-enabled computer (desktop or laptop).
* Enable webcam when prompted.
* JavaScript and WebGL must be enabled in the browser (that is a default setting).
* Check that ad blockers and firewalls do not block the webcam and other content.
* Pages containing large Wavefront files may take time to load (half to a few minutes).

Augmented reality markers

* Print on regular printer; try different sizes for different applications.
* When using Hiro and Kanji markers ensure they are printed at the same size.
* Ensure that makers have a white frame around the black drawing, at least 10% of size.
* To improve marker recognition, avoid glossy papers. Opaque paper is best.
* Lighting may also affect the quality of the AR experience.
* Markers are easier to handle if glued on solid surfaces (but avoid wrinkles).
* Cubic marker can be glued on solid rubber cube cut to appropriate size.