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

Luciano A Abriata Corresp. [1, 2]

[1] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

[2] Swiss Institute of Bioinformatics, Lausanne, Switzerland

Corresponding Author: Luciano A Abriata
Email address: luciano.abriata@epfl.ch

For years, immersive interfaces using virtual and augmented reality (AR) for molecular visualization and modeling have promised a revolution in the way how we learn, communicate and work in chemistry, structural biology and related areas. However, the tools and prototype applications available today require specialized hardware and/or software, often costly and cumbersome to setup. These limitations prevent wide reach of the technology into education and research centers in a standardized form, as well as adaptation to specific user needs. All this in turn prevents large-scale testing of the actual effects of such technologies on learning and thinking processes.Here I explore building blocks for creating AR applications that run as web pages in ordinary web browsers in regular computers equipped with webcams and internet connection, i.e. with no software/hardware impediments and at no costs, as a means to easily develop content for viewing and handling virtual molecular systems in an AR environment. I present some specific web apps of direct use, or that can be modified for further use, in teaching/learning chemistry by visualizing molecules, manually testing molecular mechanics and intermolecular interactions, etc., just like students would do with plastic molecular modeling kits but in front of a “mirror” that augments the experience with computer graphics, calculations and information overlay. I further present examples that advance applicability of these technologies to actual research, for example by simulating experimental data informative of the shape of protein-protein complexes in real time as the user moves two proteins around in space. From this playground, I then ponder how other emerging technologies could couple inside web browsers to deliver fully immersive, interactive, collaborative molecular modeling tools that can streamline human thought and intent with the numerical processing power of computers, at affordable costs.These prototypes and ideas should serve as starting points for building active content that everybody can utilize online, enabling for the first time wide reach of the tools and in turn mass-testing of the effect of immersive technologies on molecular modeling, and more

globally on education.

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 learn, communicate and work in chemistry, structural biology and related areas. However, ~~the~~ tools and prototype applications available today require specialized hardware ~~and/~~or software, often costly and cumbersome to setup. These limitations prevent wide ~~reach~~ of the technology in~~to~~ education and research centers in a standardized form, as well as adaptation to specific user needs. All this in turn prevents large-scale testing of the actual effects of such technologies on learning and thinking processes.

Here I explore building blocks for creating AR applications that run ~~as web pages~~ in ordinary web browsers ~~in~~ regular computers equipped with webcams and internet connection, ~~i.e. with no software/hardware impediments and at no costs,~~ as a means to easily develop content for viewing and handling virtual molecular systems in an AR environment. I present ~~some~~ specific web apps of direct use, ~~or~~ that can be modified for ~~further~~ use~~,~~ in ~~teaching/learning~~ ~~chemistry~~ ~~by~~ visual~~izing~~ molecules, manually test~~ing~~ molecular mechanics and intermolecular interactions, etc., just like students would do with plastic molecular modeling kits but in front of a “mirror” that augments the experience with computer graphics, calculations and information overlay. I further present examples that ~~advance~~ applicability of these technologies ~~to actual~~ research, for example by simulating experimental data informative of the shape of protein-protein complexes in real time as the user moves two proteins around in space. From this playground, I then ~~ponder~~ how other emerging technologies could ~~couple inside web browsers~~ to deliver fully immersive, interactive, collaborative molecular modeling tools ~~that can streamline human thought and intent with the numerical processing power of computers, at affordable costs~~.

These prototypes and ideas ~~should~~ serve as starting points for building active content that everybody can utilize online, enabling for the first time wide reach of the tools and in turn mass-testing of the effect of immersive technologies on molecular modeling, and more globally on 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 of science that deal with abstract objects, objects much smaller or larger than human dimensions, objects that are hard to acquire and handle due to high costs, scarcity or fragility, and very large amounts of data (O'Donoghue et al., 2010; Matthews, 2018; M. Krichenbauer et al., 2018; Sommer et al., 2018). Chemistry and structural biology are examples of such disciplines where AR and VR have been attributed high potential in education and research, by ~~conveying~~ hybrid physical/computational interfaces to handle and explore virtual molecules in real 3D space augmented with real-time overlay of information from databases and calculations. However, the actual impact of such immersive technologies on teaching, learning and also on working in chemistry still requires deep evaluation,(M. Fjeld & B. M. Voegtli, 2002; Pence, Williams & Belford, 2015; Matthews, 2018; Bach et al., 2018; Yang, Mei & Yue, 2018) which has ~~so far~~ progressed very slowly due to the complex software setups and the specialized hardware needed ~~for currently available tools,~~ which limit availability, reach, adoption, and thus testing. Indeed this limitation is shared more broadly with other potential applications of AR and VR in science, which so far "[…] remain niche tools for scientific research".(Matthews, 2018)

In the last two decades several groups have been studying ways to achieve immersive environments for chemistry and structural biology using VR and ~~also some~~ AR (Gillet et al., 2004, 2005; Maier, Tönnis & GudrunKlinker, 2009; Maier & Klinker, 2013; Hirst, Glowacki & Baaden, 2014; Berry & Board, 2014; Martínez-Hung, García-López & Escalona-Arranz, 2016; Vega Garzón, Magrini & Galembeck, 2017; Balo, Wang & Ernst, 2017; Goddard et al., 2018b,a; Wolle, Müller & Rauh, 2018; O'Connor et al., 2018; Ratamero et al., 2018; Müller et al., 2018; Stone, 2019). Such interfaces allow handling molecules over 6-degrees of freedom and with both hands, even multiple users ~~concurrently~~. ~~This is~~ expected to overcome the limitations of ~~traditional software based on~~ screen, mouse and keyboard, ~~thus~~ enabl~~ing~~ more intuitive, fluid exploration of molecular features and data. ~~Of note,~~ most of these works ~~do not yet present~~ complete AR or VR versions of fully-fledged programs, but rather prototypes, proofs of concept and case studies on how humans interact with virtual molecules in AR or VR. Notable highlights moving towards complete immersive molecular visualization and modeling programs are the VR rewrite of Chimera, ChimeraX (Goddard et al., 2018b), new VR plugins for the VMD molecular graphics program (Stone, 2019), and a few commercial programs, all specifically tailored ~~to~~ VR ~~interfaces.~~

~~Of central relevance to this article,~~ ~~m~~ost of these works suffer from two severe limitations. First, all but a few exceptions require ~~*ad hoc*~~ ~~pieces of~~ hardware such as head-mount displays (helmets, headsets or goggles like MS Hololens, Oculus Rift, HTC Vibe, etc.) or "caves" (essentially large surround screens) plus 3D-handheld input devices and the corresponding computer hardware ~~that links the visualization and handheld devices and provides computing power~~. The few remarkable exceptions are prototypes using ordinary webcam-enabled smartphones (Balo, Wang & Ernst, 2017) or laptops (Gillet et al., 2004). The second ~~strong~~ limitation ~~of several approaches~~ is that ~~they~~ require specialized programs that often must be correctly interfaced to couple the different components required for an AR/VR experience, i.e. tracking limbs, handheld devices or AR markers, then running calculations on molecular data, and finally displaying results and molecular graphics on screen (as mere examples, see (Ratamero et al., 2018) and (Gillet et al., 2005)). Some such programs are only compatible with *ad hoc* VR devices, and many are not free. Overall, then, despite the dropping costs, access to these tools still requires investments in the order of hundreds to low-thousand dollars per user, and often software interfacing that may not be at hand of lay students and teachers. Therefore, these solutions cannot yet be widely available across the world, let alone having one such device per student, and totally out of reach for educational centers in underdeveloped countries.

The second big limitation of most current setups is that they are limited to VR, but it has been shown that AR is much more adequate for educational purposes, because by not occluding the view of the user's own limbs, AR results in better motion~~al~~ coordination and object handling than VR,(Sekhavat & Zarei, 2016; M. Krichenbauer et al., 2018) which is essential from a pedagogical perspective. ~~Besides,~~ in AR the view of the world is not obstructed hence students ~~can keep fluent contact with teachers as they work~~.

The purpose of this work is to demonstrate that current client-side web technologies allow building web pages for AR-based molecular visualization, and some molecular modeling too, running just on web browsers in regular computers. This enables the easy creation of

~~pedagogical~~ immersive~~-~~material that can be employed by students and teachers at very affordable costs and with very simple setups. A~~ctually a~~ll they must do is just access a webpage, enable webcam activity in the browser, and show it a printed AR marker on which the molecules will be displayed. From the developer side, the code is very simple, to the extreme that visualization-only applications are achievable just with HTML code, while JavaScript ~~programming~~ allows incorporating interactive ways to handle molecules, change their colors and representations, compute interactions, etc.

This article begins by briefly reviewing the client-side web programming technologies that enable, in ordinary web browsers running on standard devices (i.e. computers and also tablets and smartphones, although the latter are not yet smoothly supported), developments of graphical interfaces for AR rich gesture- and speech-based commands, in-browser calculations for molecular mechanics and for comparison to experimental observables, concurrent collaboration through the world wide web, and other human-computer interaction technologies. In a second section the article showcases prototype web apps for specific tasks of practical utility in pedagogical settings on chemistry, biophysics and structural biology, and even some early examples that ~~advance a~~ potential for these web apps in assisting ~~actual~~ research. Being based on ~~fully~~ open technolog~~ies~~ that require ~~no installations and run on~~ modern browser~~-enabled computer~~ “out of the box”, educators and researchers are free to try out all these examples right away~~,~~ following the links in the figure captions (~~many accompanied by~~ online videos) and encouraged to start building their own web apps profiting from the intrinsic simplicity and modularity of the approach.

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

Virtual and augmented reality

At the core of immersive experiences ~~there is~~ visualization based on virtual or augmented reality methods (VR and AR). VR is the total replacement of the real world by a simulated environment, whereas AR is the superposition of virtual objects on top of the real world. While VR is probably experienced best with VR goggles to suppress any side view of the real world, AR is amenable to other kinds of less immersive devices like desktop or laptop computers, tablets and smartphones, making it better suited for commodity solutions. On top, AR seems to be perceived by humans more naturally than VR (Sekhavat & Zarei, 2016; Y. Gaffary et al., 2017; M. Krichenbauer et al., 2018).

The focus of this article is on the development of AR-based web pages for ordinary computers equipped with a frontal webcam (as in most regular laptops). Such setup function~~s~~ as an “augmented reality mirror” where the user sees him/herself with the virtual objects, here molecules, in the hands (Figure 1A-C). These web apps work “out of the box” in the most popular web browsers, ~~without need of any software installation or manual updates (at least with browsers as installed and with default settings,~~ details below and troubleshooting in Table 1). In principle it is also possible to write web pages that allow for look-through AR in tablets and smartphones, or VR in smartphones plugged into cardboard goggles with proper eye lenses for stereoscopy; however, these do not yet run seamlessly and “out of the box” across different devices –although future built-in software for VR could open up this possibility.

In web browsers, the WebGL API provides powerful 2D and 3D graphing capabilities (using GPU resources) in a format fully integrable with other web standards, HTML elements, APIs and JavaScript libraries, without the need of plug-ins and highly standardized across browsers. It is thus possible to couple all elements required to build up an AR experience directly inside the web page’s code. A handful of JavaScript libraries exploit WebGL to facilitate scene rendering, Three.js being probably the most widely used. In turn, tools like A-Frame provide entity component system frameworks that wrap Three.js into HTML language tags for building AR and VR scenes very easily. The examples presented here showcase either direct use of Three.js or Three.js through A-Frame for AR. These libraries/frameworks can be used either by (i) loading pre-made models of the molecular systems in formats like Wavefront ~~(~~which can be exported ~~as OBJ+MTL files~~ straight out of molecular visualization programs like VMD (Humphrey, Dalke & Schulten, 1996), UnityMol (S. Doutreligne et al., 2014; Wiebrands et al., 2018), ChimeraX (Goddard et al., 2018b), etc. as in the example in Figure 1C, possibly further edited with Blender as in (Martínez-Hung, García-López & Escalona-Arranz, 2016)); or (ii) employing 3D primitives (like spheres, cylinders, etc.) to draw the molecular systems from scratch up from atomic coordinates (Figure 1A,B and most other examples in this article). Use of ~~ready~~ Wavefront models is much simpler (~~requiring~~ only a few lines of HTML code to load and display objects); while ~~use of~~ 3D primitives ~~needs~~ larger pieces of code to map all 3D objects from atomic coordinates to their display positions, but in turn allows for much finer control of shapes, sizes, colors, and positions, which are key to interactivity.

Looking into the future, web browser developers are working on directly integrating AR/VR as a native API, which should provide smoother and higher resolution graphics; however this is largely experimental at the moment (Medley, 2019).

Object detection and tracking

The other key component required for AR and VR is a means to detect and track objects or parts of the user's body such as the hands, in order to manipulate virtual objects. Applications using *ad hoc* hardware use sensors and cameras that track the user's position in space and hand-held controllers, usually seen by the user as virtual tweezers or hands, to directly move objects in space. For commodity AR/VR in web browsers, solutions rely on computer vision through the webcam, essentially tracking algorithms like ARToolKit's JavaScript version (Kato, 1999) among other similar solutions. This library essentially tracks user-defined 2D markers (examples in Figure 2A and in S1 ready to print at different sizes) in space as long as they are visible to the webcam, and make their computed coordinates available to the graphics algorithms. One particularly interesting implementation is AR.js and its A-Frame wrap, which enable highly simplified AR/VR, even using exclusively HTML code for simple developments (for example, in Figure 1C).

It is important to note that in marker-based AR different viewers receive different perspectives of the same marker and hence of the rendered virtual object, just as if it was a real object in real space (Figure 2B). This easily enables multi-user AR in a common room, as would be useful in a classroom setting where students and teachers look at the same virtual molecule.

An alternative to traditional marker-based AR should in principle be possible by using a plain hand-tracking JavaScript library like Handtracking.js. Another slightly more expensive approach but possibly better in tracking performance is using a device like the infrared-based Leap Motion Controller, which includes a JavaScript library to feed positional information from the device into the web app. Unfortunately, however, there are currently no ready-to-use libraries that couple these input tools to WebGL graphics, so these avenues would require further developments before they can be used.

Current JavaScript libraries for computer vision allow even more complex object tracking. One interesting example is gesture recognition by the WebGazer.js library, which analyzes face features to estimate where on the screen the user is looking at (Papoutsaki et al., 2016). In molecular visualization this can be used for example to automatically move regions of interest to the front of the field of view, as in the example in Figure 2C.

Speech-based interfaces

On top of gesture recognition, a speech-based interface can be highly useful for situations in which the user's hands are busy holding objects, as would happen very often in AR/VR applications. Current in-browser speech recognition APIs enable this very easily, especially through libraries like Annyang (Ater, 2019) used in some of the examples of this article. These libraries usually allow working in two modes, one where the browser waits for specific commands (most accept variable inputs) and one where the browser collects large amounts of text that are then made available to the environment. The former allows direct activation of functions without the need for the user to click on the screen. The second option opens up the possibility of automatically detecting subjects, actions and concepts that are fed to artificial intelligence, or just predefined rules, that the computer will analyze in background. For example, when two users are discussing the interaction surface between two proteins and mention certain residues, the computer could "understand" this and automatically mine PubMed for mentions of said residue. This may seem far-fetched, but is essentially the same technology that underlies automatic advertising and suggestions based on users' various inputs and usage statistics in software and web sites. In turn, the problem of intelligently suggesting chemical and biological information related to a topic or object has already been addressed for some time, for example in information augmentation tools like Reflect (Pafilis et al., 2009), advanced text-mining tools (Rebholz-Schuhmann, Oellrich & Hoehndorf, 2012), the evolution of science-related standards, formats and content for the semantic web (Hendler, 2003), and of course thanks to ever-increasing scientific databases and ontologies, which standardize knowledge and data.

Intensive calculations

As reviewed in a recent special issue of *Computing and Science in Engineering* (M. DiPierro, 2018), JavaScript has become a very powerful language through several means, including the creation of language subsets specialized for speed, optimized just-in-time compilation, methods to program background scripts, and libraries to perform multi-core and on-GPUs computing to accelerate intensive calculations. It is in fact now possible to trans~~com~~pile ~~from~~ C/C++ and other languages directly ~~in~~to JavaScript retaining close-to-native execution speeds, allowing web browsers to support quite complex data analysis and visualization ~~coded~~ directly in web pages (Abriata et al., 2017; Abriata, 2017a). Possible applications of this power related to the development of interactive, immersive systems for molecular modeling include simulating molecular mechanics and experimental data, performing numerical data analysis and even handling data in neural networks, etc. If execution speeds get fast enough, these possibilities enable applications where the user gets real-time numerical response from the web app as (s)he manipulates the molecular systems. Some of the prototypes presented in the second section of this article include such examples.

Besides the specific calculations that may be coded *ad hoc* for certain problems, there are several libraries that can be of use, saving large writing time and often being developed by specialists. While a large list is reviewed in (Abriata, 2017a) and kept updated at https://lucianoabriata.altervista.org/jsinscience/, I mention here one particularly useful library called Cannon.js. This JavaScript library simulates rigid body mechanics and integrates smoothly with Three.js and A-Frame, as in the example from Figure 1B where it is used to simulate thermal motions. Although rigid-body physics might look limiting for molecular modeling applications, they ~~are often enough~~ in settings requiring coarse-grain modeling like in integrative modeling; in fact the Integrative Modeling Platform contains one such kind of engine for certain applications (Russel et al., 2012). Nevertheless, implementation of more complex force fields is certainly possible, as exemplified by a JavaScript trans~~com~~pilation of the OpenMD engine (Jiang & Jin, 2017).

Further building blocks

Any other technology that facilitates interaction with the computer within a 3D environment, either to deliver or obtain information, might be of use. For example, haptic feedback would be desirable to confer a physical feel of interactions and clashes as in (Wollacott & Merz, 2007; Stocks, Hayward & Laycock, 2009; Stocks, Laycock & Hayward, 2011; Matthews et al., 2019). Achieving a good experience in haptic feedback requires specialized devices and is ~~currently~~ an area of active research (Bolopion et al., 2009, 2010), therefore it ~~is~~ clearly not ~~applicable easily and inexpensively to in-browser programs for the masses~~ (although possibly some less expensive alternative based on commodity robotics could be explored). Other rudimentary ways to achieve sensory feedback include exploiting built-in vibration devices and touch-pressure sensing in touch screens, both handled by JavaScript APIs.

A particularly interesting aspect of software running on web browsers is the ease with which different users can connect to each other, just over the internet. Web apps can exploit browser communication sockets to achieve browser-to-browser links over which data can be transmitted freely, with a server only intervening to establish the initial connection. For example, two or more users can concurrently work on a JSmol session by sharing mouse rotations and commands on all other users' screens with a few milliseconds of delay (because only instructions are transmitted, not graphics) (Abriata, 2017b). Such technology could be adapted to complex immersive environments to allow multiple users to work on the same problem at a distance, essential for collaboration, demonstrations, and online teaching (Lee, Kim & Kang, 2012).

Part 2: Prototype web apps showcasing sample applications

This section presents example AR web apps compatible with major web browsers in modern computers, introducing features of increasing complexity. All examples are accessible through links in the Figure captions and in a list maintained at http://lucianoabriata.altervista.org/papersdata/tablepeerjcs2019.html, which also contains additional examples. Figure captions and the online list also point at online videos where the examples are shown at work. To run these AR web apps the user needs to print the Hiro, Kanji or cube markers at the same scale (Figure 2A and S1, and links on web pages). For simpler handling, flat markers (Hiro and Kanji) may be glued on a flat surface mounted on a device that can be easily rotated from the back, such as a small shaft perpendicular to the marker plane. The cube marker (used in Figure 1B) is printed in a single page that then needs to be folded and possibly glued to a solid cube made of wood, plastic, rubber or similar material.

Introducing web browser-based AR

The simplest web setup for AR consists in using A-Frame to either (i) display a representation exported from VMD (Humphrey, Dalke & Schulten, 1996) in Wavefront (OBJ/MTL) format on a marker (example in Figure 1C); or (ii) display primitive spheres at coordinates computed from atomic positions and colored by atom type (similar to the example in Figure 1A). Since only A-Frame is used in such examples, only HTML tags are needed, which the user can inspect in the source code of the pages listed in the caption to Figure 1 (or in the HTML section of the code shown in Figure S2).

Web apps using A-Frame can gain interactivity through portions of JavaScript code that read atom coordinates and carry out calculations on them. In the example shown in Figure 1A the user drives a lysine side chain with the Hiro marker and a glutamate side chain with the Kanji marker. Each molecule is anchored to the center of its corresponding AR marker through its CA atom. Their protonation states correspond to neutral pH, so lysine is protonated hence its N atom (blue) is charged by +1, whereas glutamate is deprotonated hence its O atoms (red) bear a total charge of -1. Through simple JavaScript code the web app (i) ~~adapts the positions of~~ 10 yellow spheres ~~that~~ connect ~~in space~~ the lysine's N atom ~~with~~ one of the glutamate's O atoms; (ii) reports the distance between these 2 atoms and the corresponding attractive electrostatic force in real time; and (iii) checks for and displays clashes between any pair of atoms of the two molecules. The code for (i) is wrapped inside an auxiliary .js file, and the code for (ii) and (iii) is located between <script> tags at the end of the HTML file. The distance, electrostatic and clash calculations are computed inside a setInterval() function that is executed every 200 ms and uses the id ~~identifiers~~ of the sphere tags to locate them (see code in Figure S2). The distance is calculated in Angstrom and includes a correction for a zoom factor that scales atom sizes and positions when the molecular coordinates are parsed into A-Frame HTML, to properly fit the screen. Clashes are detected ~~as~~ two spheres ~~being~~ within 3 Å and displayed as semitransparent A-Frame spheres centered on the affected atoms.

All the above examples use a single AR marker per molecule, which is easy to setup but is limited to orientations where the marker is visible. With multiple coupled markers, for example a cube ~~made of~~ 6 different markers ~~in~~ its faces, the user can freely rotate the object in 3D space, with the additional advantage that the estimation of position and orientation works better. One such example is shown in Figure 1B, from a web app that allows manipulation of any molecule loaded in PDB format.

Adding interactivity: examples on small molecules

An interesting modification of the example using lysine and glutamate side chains is incorporating a very simplistic emulation of hydrogen bond detection and proton transfer. Here, JavaScript code calculates and displays hydrogen bonds when the geometry permits, and randomly “transfers” one proton from lysine’s N atom to one of the O atoms of glutamate if the lysine is protonated, or the other way around (actually spheres attached to each marker are hidden or shown as needed to emulate proton transfer). Protons “jump” only when they are within 2 Å of the receiving N or O atom; and they are set to jump back and forth to reflect 70% time-averaged population of protonated lysine and 30% of protonated glutamate, to convey the feeling of different acidic constants (in this case set to 70/30 = 2.33 favoring protonated lysine, shifted orders of magnitude from the real acidic constant to observe protonation of glutamate during sensible periods of time). At distances longer than 2 Å but shorter than 3 Å the web app displays a yellow dotted line that represents a hydrogen bond between the potential receiver heavy atom and the involved proton. See this example at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/smallmolprotontransfer.html .

Similar emulation strategies could be easily used to build “interactive animations” for exploring chemical and physical phenomena of much pedagogical use, as in the PhET interactive simulations but using AR to directly, almost tangibly, handle molecules.~~(Moore et al., 2014)~~For example, Figure 3A illustrates stereoselectivity in the Diels-Alder reaction in interactive 3D. This reaction occurs between a dienophile and a conjugated diene in a concerted fashion, such that the side of the diene where the initial approach occurs defines the stereochemistry of the product. The web app in this example allows users to visualize this in 3D as they approach a molecule of 1,3-cyclohexadiene held with the left hand (Hiro marker) and a molecule of vinyl chloride in the right hand (Kanji marker). As the two pairs of reacting C atoms approach simultaneously, the two new bonds gain opacity until the product is formed. Additionally, the product formed in this reaction is by itself an interesting molecule to visualize and move around in 3D through AR, because it contains two fused six-membered rings which are often hard to understand in 2D.

Looking into the near future, there is a very important point regarding the idea of emulating reactivity as in the proton transfer and Diels-Alder reaction examples. These examples are merely pictorial visualizations of the mechanisms, and ~~they are~~ not based on any kind of quantum calculations. Such calculations are extremely slow to be incorporated into immersive experiences where energies need to be computed on the fly. However, novel machine learning methods that approximate quantum calculations through orders-of-magnitude faster computations (like in (Paruzzo et al.; Smith, Isayev & Roitberg, 2017; Bartók et al., 2017)) could in a near future be coupled to AR/VR systems to interactively explore reactivity ~~with realistic energy profiles~~ in real time. Such tools would be useful not only for education but also for actual research, for example to interactively test the effect of chemical substituents on a reaction, estimate effects on spectroscopic observables, probe effects of structural changes on molecular orbitals, etc.

Another interesting tool to integrate with AR/VR is a physics engine, to add realistic mechanics to the simulation. The web app in Figure 1B uses Cannon.js to simulate thermal motions and thus give a sense of dynamics to the visualized system. And as presented in the first section ~~of the article~~, it is possible to ~~adapt~~ ~~more complex~~ force fields better suited for molecular interactions ~~in~~to JavaScript ~~versions~~ (Jiang & Jin, 2017).

~~Possible applications go well beyond emulating reactivity and dynamics, being especially interesting the possibility of using JavaScript to carry out~~ on-the-fly calculations of experimental observables ~~that~~ can be compared in real time to ~~a piece of actual~~ experimental data, or simply to explore possible outcomes of an experiment. Figure 3B exemplifies this with the calculation of paramagnetically induced pseudocontact chemical shift and line broadening on a probe atom attached to one AR marker, as it is moved around the heme group of metmyoglobin with the other AR marker. This web app implements standard equations from the theory of paramagnetic nuclear magnetic resonance (Bertini, Turano & Vila, 1993) fed on-the-fly with the corresponding polar coordinates, then simulates the spectrum including noise and displays it using the Google Charts JavaScript library. More examples with realistic applications of client-side calculations in molecular modeling are introduced below, ~~for biological macromolecules~~.

AR-based visualization and modeling of biological macromolecules

Figure 1C already exemplified a VMD session directly imported into AR.js-A-Frame, for a very simple visualization-only AR experience in web browsers. Figure 4A does the same on a VMD scene reproducing the work of Berry and Board(Berry & Board, 2014) ~~but achievable~~ with a web browser *i.e.* requiring no program, plug-ins nor driver installation, and accessible just on a web page. But besides using premade VMD scenes, the developer can also convert atomic coordinates to Three.js or A-Frame spheres as in the helical model of a nuclear export signal from~~(~~Bologna et al.~~,~~ 2018) (Figure 4B) which ~~readily~~ shows the functionally relevant amphipathic nature of these protein motifs. Moreover, with the web app depicted in Figure 1B one can load any file from the Protein Data Bank (although currently only small proteins are well supported) to look at its contents in AR, and even emulate thermal motions and protein unfolding at high temperatures through Cannon.js.

The next examples showcase the prototype incorporation of restraints, primitive force fields and on-the-fly simulation of experimental observables in web browser-based AR for molecular modeling.

The example in Figure 5A allows driving in space two molecules that are known to form a complex in solution, specifically ubiquitin (red trace) and a ubiquitin-interacting motif (UIM, blue trace) taken from PDB ID 2D3G(Hirano et al., 2006). The web app simulates on-the-fly the small-angle X-ray scattering (SAXS) profiles expected from the relative arrangement of the two proteins~~, overlaying~~ simulated profiles ~~over~~ an experimental profile in real time as the user moves the proteins around. This offers a way to interactively test possible docking poses ~~that are compatible~~ with the experimental data. Although this of course could never be better than the extensive sampling achievable with molecular simulations, such an interactive tool could be useful (i) for preliminary analysis of SAXS data before starting complex calculations and (ii) to ~~judiciously analyze~~ the results from such calculations. ~~[~~For simplicity and speed, in this example the SAXS profile calculation is based on the Debye formula (Debye, 1915) iterated through pairs of residues instead of through all pairs of atoms as the full equation requires; however, realistic SAXS profiles of actual utility in modeling can be achieved with similar strategies and proper parameterization of the scattering centers (Stovgaard et al., 2010)~~]~~. This web app further includes a rudimentary residue-grained force field (i.e. describing each amino acid with one backbone and one side chain bead) to detect clashes, and a predefined binding coordinate which upon activation brings the two molecules together. Activation of SAXS simulation, clash-detecting force field and binding coordinate are controlled by voice commands, required because the user's hands are busy handing the markers. This proceeds through the browser's speech recognition API ~~and~~ does not consume much resources ~~because speech recognition occurs in the cloud~~. The ~~possibility of incorporating~~ all these different elements (AR, 3D visualization, calculations and speech recognition) illustrates the superb integration capability of libraries for client-side scripting in web browsers (Abriata, 2017a). ~~As a last note regarding this example,~~ the modularity and simplicity of client-side web programming allows easy adaptation to other kinds of experimental ~~data, as in the~~ example on residue-specific paramagnetic relaxation enhancements by another researcher at https://rrasia.altervista.org/HYL1_1-2/Hyl1_12_minima.html

~~Another example, presented in~~ Figure 5B~~,~~ shows how AR can help to explore ~~residue-~~residue contact ~~data~~. This information essentially points at pairs of residues ~~that~~ are in contact (Simkovic et al., 2017), either inside folded proteins (Abriata et al., 2018) or, more relevant to this example, in protein-protein complexes (Ovchinnikov, Kamisetty & Baker, 2014). The example shows contacts predicted from coevolution analysis of large sequence alignments for the pair of proteins in chains A and B of PDB entry 1QOP, taken from the Gremlin server (Ovchinnikov, Kamisetty & Baker, 2014) at http://gremlin.bakerlab.org/cplx.php?uni_a=1QOP_A&uni_b=1QOP_B. Each protein is driven by one marker, and the predicted contacts are overlaid as dotted lines connecting the intervening pairs of residues. These lines are colored green, olive and red according to decreasing coevolution score as in the Gremlin website, and their widths reflect in real time the distance between pairs of residues, presumably minimal when contacts are satisfied if the prediction is true. Such kind of tool could allow interactive inspection of residue-residue contact data to actively detect false positives through human intervention, to be removed before running restraint-guided docking.

The last prototype application shows rudimentary handling of highly disordered protein regions, in this case to test how a linker made of 6 glycine ~~(which are very flexible)~~ residues enables space exploration of two well-folded domains ~~joined by the flexible linker~~ (Figure 6). The example considers only the dynamics of very flexible linkers within the very coarse approximation that beads describing amino acids behave as rigid spheres connected by fixed distances, obtained as average distances between consecutive pairs of alpha carbons. Despite ~~very~~ simpli~~stic,~~ such rigid-body-like descriptions are often sufficient for describing several aspects of biological matter, especially when docking large macromolecules in, for example, the initial stages of integrative modeling approaches (Russel et al., 2012).

More specifically, the example in Figure 6 handles flexibility through Cannon.js, presented earlier for the example in Figure 1B but here wrapped into HTML through the A-

Frame-physics extension. Each globular domain (one corresponding to ubiquitin and the other to the interacting helix) is modelled with 2 to 4 rigid beads per residue, ~~*i.e.* a description similar to that of the MARTINI force field(Marrink et al., 2007)~~, while the flexible linker is modelled as backbone beads only (similar to the MARTINI description for glycine). As the user moves the two domains around, one attached to each AR marker, the web app updates the positions of the residues of the flexible linker based on the Cannon.js force field. This very simple model can help ~~to~~ answer questions related to the separation of the anchor points and the maximal extension of the linker when straight: How far can the two proteins go with the given linker containing 6 residues? Can the interacting partner be docked through this or that interface yet allow for a relaxed configuration of the linker? ~~Such~~ investigations are ~~in turn~~ assisted by on-the-fly estimation of entropy associated to given extensions of the linkers, estimated ~~here from a~~ worm-like chain model from polymer physics (Marko & Siggia, 1995), and by the strain on the linkers ~~which essentially tracks how much the consecutive glycines are stretched~~ when the user pulls them apart beyond their equilibrium distance.

Discussion

Achieving smooth and useful immersive chemistry is one of the key “grand challenges” for the simulation of matter in the 21st century (Aspuru-Guzik, Lindh & Reiher, 2018), which also applies to multiple other disciplines where interactive computer interfaces are expected, but yet not proved, to help to grasp and explore abstract concepts and objects. Heading in this direction, the last two decades saw a large number of works introducing different ways to achieve AR and VR for chemistry and structural biology~~, with some of the most important contributions highlighted in the Introduction section.~~ So far, however, ~~all~~ these works fall short in enabling wide use of the tools due to hardware and software costs ~~and~~/or ~~to the need~~ of software setups. In contrast, the building blocks put forward in this work should easily allow building tools that can permeate widely into educational institutions and homes, in wealthy and in underdeveloped countries as well. I have verified that, when running on laptop and desktop computers with several flavors of Linux, MacOS and Windows, the web apps run out-of-the-box. This means that these web apps become, upon upload to a server, immediately accessible on a laptop or desktop computer (not so in smartphones or tablets) with no need for plugins, no installations, no manual updates, and no associated costs. In fact, the reader could right now print AR markers, access the webpages indicated in the figure captions, and test the examples right away.

Like most other works, this one presents proof~~s~~-of-concept and prototypes, but does not provide the complete set of features of a full program like ChimeraX~~ and some commercial alternatives. On the good side~~ compared to other prototypes, and also to the full programs, the hardware and software setups as well as the code required to create new examples are very simple. This should allow for wide use in educational settings, even at one device per student if students can bring their own laptops to the school, and even running examples at their homes. In education, such tools could replace/complement physical (like plastic-made) modeling kits, allowing virtually unlimited numbers and kinds of atoms, as well as augmenting models with additional information such as forces, charges, electron clouds and orbitals, data facts,~~ etc~~. In research, such tools could help to visualize and probe molecular structure, simulate expected outcomes of experiments and test models and simulated data against experimental data, ~~etc.~~, all through intuitive cues and fluent human-computer interactions.

Conclusions and Future Perspectives

Now that the core elements have been highlighted and proofs-of-concept set, the next stage in the quest for ~~free~~, ~~world-wide~~ web apps for education is to develop actual content of pedagogical value for teachers and students. The simplest ~~such web apps~~ could ~~consist in~~ mere visualization to explore molecular geometries, surfaces, orbitals, etc., with specific sets of demos to assist learning of concepts ~~about~~ chirality, organic molecules, metal coordination complexes, biomacromolecular structures, just to mention a few obviously interesting cases. By adding mechanics more complex demos could be created where students can ~~for example~~ interactively explore torsion angles as in the textbook case of probing the energy landscape of rotation around the central C-C bond of butane, or swap between chair and boat conformations of 6-member rings, explore hydrogen bonding patterns between say opposing beta strands facing each other in a protein or in a protein complex, etc. Importantly, every single student having a computer at hand could use these apps, not only at the school/university but also at home, therefore this could become an actual learning tool of full-time, individual use. The possibility of reaching the masses with this kind of web technologies for AR-based molecular modeling in turn opens up the possibility of performing large-scale evaluations of their actual impact in education.

Moving next into the future, I believe it is worth exploring if full working web programs for AR-based molecular modeling may actually become powerful enough to also assist research. Some of the prototypes presented here advance possible uses, by allowing for example simulation of data protein-protein docking poses and comparison to the corresponding experimental data in real time (SAXS example in Figure 5). Some issues that should ideally be addressed before creating fully-fledged web programs useful for research beyond education include (i) improving AR marker detection and tracking to stabilize inputs, (ii) developing some kind of AR marker that is clickable so that users can drag and drop objects in space, this possibly achievable through commodity robotics, (iii) improving graphics, where the possibility of adapting existing web molecular graphics like NGL (Rose & Hildebrand, 2015), 3dmol (Rego & Koes, 2015), Litemol (Sehnal et al., 2017), Mol*(Sehnal et al., 2018), JSmol (Hanson et al., 2013), etc. is particularly enticing, and (iv) developing force fields that correctly handle molecular dynamics and energetics for different tasks, which may imply different degrees of graining for different applications. Another global improvement, also important for pedagogical applications, would be incorporating proper object occlusion, which is still non-trivial and subject of studies in the AR community.

Some further directions that could be explored in the near future are fluently connecting through an AR experience users located physically distant, so that they can collaborate on a research project or teach/learn at the distance. Adapting future standards for commodity AR/VR in smartphones (plugged into cardboard goggles for stereoscopy) is also worth exploring as this would lead to an even more immersive experience than the mirror-based apps proposed here. Note that smartphones may be limited in power so this will further require coupling the web apps to external computer power. Last, pushing the limits towards fully immersive visual analytics for molecular modeling, and especially thinking about solutions useful for research, I can forecast the addition of support for haptic devices for force feedback, AR without markers i.e. just by tracking the users' hands and fingers, and capability to respond to active verbal commands and to other visual or audio keywords taken from the context by automatically checking databases and mining literature to propose relevant pieces of information.

Acknowledgements

I acknowledge all the members of the communities that develop the client-side web programming tools used here, as well as the very helpful communities who contribute to their improvement, bug detection and correction, documentation, and online help. Special thanks to J. Ethienne (AR.js developer), D. McCurdy (Google), T. Ater (Annyang developer), L. Stemkoski (Adelphi U., USA), A. Herráez (U. Alcalá, Spain) and A. Papoutsaki (Pomona College, USA). I also acknowledge numerous researchers from the physics, chemistry and biology communities who have provided ideas, suggestions, and support, especially L. Krapp, S. Träger, M. Dal Peraro, A. Barducci, R. Rasia, G. van der Goot and M.E. Zaballa. I also greatly acknowledge the extremely helpful revisions from the Editor and two reviewers of this work.

References

Abriata LA. 2017a. Web Apps Come of Age for Molecular Sciences. *Informatics* 4:28. DOI: 10.3390/informatics4030028.

Abriata LA. 2017b. Concurrent interactive visualization and handling of molecular structures over the Internet in web browsers. *ArXiv e-prints*.

Abriata LA, Rodrigues JPGLM, Salathé M, Patiny L. 2017. Augmenting research, education and outreach with client-side web programming. *Trends in Biotechnology*.

Abriata LA, Tamo G, Monastyrskyy M, Kryshtafovych A, Dal Peraro M. 2018. Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods. *Proteins: Structure, Function, and Bioinformatics* 86 Suppl 1:97–112.

Aspuru-Guzik A, Lindh R, Reiher M. 2018. The Matter Simulation (R)evolution. *ACS Central Science* 4:144–152. DOI: 10.1021/acscentsci.7b00550.

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

Bach B, Sicat R, Beyer J, Cordeil M, Pfister H. 2018. The Hologram in My Hand: How Effective is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented Reality? *IEEE transactions on visualization and computer graphics* 24:457–467. DOI: 10.1109/TVCG.2017.2745941.

Balo AR, Wang M, Ernst OP. 2017. Accessible virtual reality of biomolecular structural models using the Autodesk Molecule Viewer. *Nature Methods* 14:1122–1123. DOI: 10.1038/nmeth.4506.

Bartók AP, De S, Poelking C, Bernstein N, Kermode JR, Csányi G, Ceriotti M. 2017. Machine learning unifies the modeling of materials and molecules. *Science Advances* 3. DOI: 10.1126/sciadv.1701816.

Berry C, Board J. 2014. A Protein in the palm of your hand through augmented reality. *Biochemistry and Molecular Biology Education: A Bimonthly Publication of the International Union of Biochemistry and Molecular Biology* 42:446–449. DOI: 10.1002/bmb.20805.

Bertini I, Turano P, Vila AJ. 1993. NMR of paramagnetic metalloproteins. *Chem.Rev.* 93:2833–2932.

Bologna NG, Iselin R, Abriata LA, Sarazin A, Pumplin N, Jay F, Grentzinger T, Dal Peraro M, Voinnet O. 2018. Nucleo-cytosolic Shuttling of ARGONAUTE1 Prompts a Revised Model of the Plant MicroRNA Pathway. *Molecular Cell* 69:709–719.e5. DOI: 10.1016/j.molcel.2018.01.007.

Bolopion A, Cagneau B, Redon S, Régnier S. 2009. *Haptic feedback for molecular simulation*. DOI: 10.1109/IROS.2009.5354256.

Bolopion A, Cagneau B, Redon S, Régnier S. 2010. Comparing position and force control for interactive molecular simulators with haptic feedback. *Journal of Molecular Graphics & Modelling* 29:280–289. DOI: 10.1016/j.jmgm.2010.06.003.

Debye P. 1915. Zerstreuung von Röntgenstrahlen. *Annalen der Physik* 351:809–823. DOI: 10.1002/andp.19153510606.

Gillet A, Sanner M, Stoffler D, Goodsell D, Olson A. 2004. Augmented reality with tangible auto-fabricated models for molecular biology applications.

Gillet A, Sanner M, Stoffler D, Olson A. 2005. Tangible interfaces for structural molecular biology. *Structure (London, England: 1993)* 13:483–491. DOI: 10.1016/j.str.2005.01.009.

Goddard TD, Brilliant AA, Skillman TL, Vergenz S, Tyrwhitt-Drake J, Meng EC, Ferrin TE. 2018a. Molecular Visualization on the Holodeck. *Journal of Molecular Biology*. DOI: 10.1016/j.jmb.2018.06.040.

Goddard TD, Huang CC, Meng EC, Pettersen EF, Couch GS, Morris JH, Ferrin TE. 2018b. UCSF ChimeraX: Meeting modern challenges in visualization and analysis. *Protein Science: A Publication of the Protein Society* 27:14–25. DOI: 10.1002/pro.3235.

Hanson RM, Prilusky J, Renjian Z, Nakane T, Sussman JL. 2013. JSmol and the Next-Generation Web-Based Representation of 3D Molecular Structure as Applied to Proteopedia. *Israel Journal of Chemistry* 53:207–216. DOI: 10.1002/ijch.201300024.

Hendler J. 2003. Science and the Semantic Web. *Science* 299:520. DOI: 10.1126/science.1078874.

Hirano S, Kawasaki M, Ura H, Kato R, Raiborg C, Stenmark H, Wakatsuki S. 2006. Double-sided ubiquitin binding of Hrs-UIM in endosomal protein sorting. *Nature Structural & Molecular Biology* 13:272–277. DOI: 10.1038/nsmb1051.

Hirst JD, Glowacki DR, Baaden M. 2014. Molecular simulations and visualization: introduction and overview. *Faraday Discuss.* 169:9–22. DOI: 10.1039/C4FD90024C.

Humphrey W, Dalke A, Schulten K. 1996. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics* 14:33–38.

Jiang C, Jin X. 2017. Quick Way to Port Existing C/C++ Chemoinformatics Toolkits to the Web Using Emscripten. *Journal of Chemical Information and Modeling*. DOI: 10.1021/acs.jcim.7b00434.

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

Lee J, Kim J-I, Kang L-W. 2012. A Collaborative Molecular Modeling Environment Using a Virtual Tunneling Service. *Journal of Biomedicine and Biotechnology* 2012:7.

M. DiPierro. 2018. The Rise of JavaScript. *Computing in Science & Engineering* 20:9–10. DOI: 10.1109/MCSE.2018.011111120.

M. Fjeld, B. M. Voegtli. 2002. Augmented Chemistry: an interactive educational workbench. In: *Proceedings. International Symposium on Mixed and Augmented Reality*. 259–321. DOI: 10.1109/ISMAR.2002.1115100.

M. Krichenbauer, G. Yamamoto, T. Taketom, C. Sandor, H. Kato. 2018. Augmented Reality versus Virtual Reality for 3D Object Manipulation. *IEEE Transactions on Visualization and Computer Graphics* 24:1038–1048. DOI: 10.1109/TVCG.2017.2658570.

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

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

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

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

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

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

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

Medley J. 2019. Augmented reality for the web.

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

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

O'Connor M, Deeks HM, Dawn E, Metatla O, Roudaut A, Sutton M, Thomas LM, Glowacki BR, Sage R, Tew P, Wonnacott M, Bates P, Mulholland AJ, Glowacki DR. 2018. Sampling molecular conformations and dynamics in a multiuser virtual reality framework. *Science Advances* 4:eaat2731. DOI: 10.1126/sciadv.aat2731.

O'Donoghue SI, Gavin A-C, Gehlenborg N, Goodsell DS, Hériché J-K, Nielsen CB, North C, Olson AJ, Procter JB, Shattuck DW, Walter T, Wong B. 2010. Visualizing biological data-now and in the future. *Nature Methods* 7:S2-4. DOI: 10.1038/nmeth.f.301.

Ovchinnikov S, Kamisetty H, Baker D. 2014. Robust and accurate prediction of residue-residue interactions across protein interfaces using evolutionary information. *eLife* 3:e02030. DOI: 10.7554/eLife.02030.

Pafilis E, O'Donoghue SI, Jensen LJ, Horn H, Kuhn M, Brown NP, Schneider R. 2009. Reflect: augmented browsing for the life scientist. *Nature Biotechnology* 27:508–510. DOI: 10.1038/nbt0609-508.

Papoutsaki A, Sangkloy P, Laskey J, Daskalova N, Huang J, Hays J. 2016. WebGazer: Scalable Webcam Eye Tracking Using User Interactions. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 3839–3845.

Paruzzo FM, Hofstetter A, Musil F, De S, Ceriotti M, Emsley L. Chemical Shifts in Molecular Solids by Machine Learning. *ArXiv* 1805:11541.

Pence HE, Williams AJ, Belford RE. 2015. New Tools and Challenges for Chemical Education: Mobile Learning, Augmented Reality, and Distributed Cognition in the Dawn of the Social and Semantic Web. In: *Chemistry Education*. Wiley-VCH Verlag GmbH & Co. KGaA, 693–734. DOI: 10.1002/9783527679300.ch28.

Ratamero EM, Bellini D, Dowson CG, Römer RA. 2018. Touching proteins with virtual bare hands : Visualizing protein-drug complexes and their dynamics in self-made virtual reality using gaming hardware. *Journal of Computer-Aided Molecular Design* 32:703–709. DOI: 10.1007/s10822-018-0123-0.

Rebholz-Schuhmann D, Oellrich A, Hoehndorf R. 2012. Text-mining solutions for biomedical research: enabling integrative biology. *Nature Reviews. Genetics* 13:829–839. DOI: 10.1038/nrg3337.

Rego N, Koes D. 2015. 3Dmol.js: molecular visualization with WebGL. *Bioinformatics (Oxford, England)* 31:1322–1324. DOI: 10.1093/bioinformatics/btu829.

Rose AS, Hildebrand PW. 2015. NGL Viewer: a web application for molecular visualization. *Nucleic Acids Research* 43:W576-579. DOI: 10.1093/nar/gkv402.

Russel D, Lasker K, Webb B, Velázquez-Muriel J, Tjioe E, Schneidman-Duhovny D, Peterson B, Sali A. 2012. Putting the pieces together: integrative modeling platform software for structure determination of macromolecular assemblies. *PLoS biology* 10:e1001244. DOI: 10.1371/journal.pbio.1001244.

S. Doutreligne, T. Cragnolini, S. Pasquali, P. Derreumaux, M. Baaden. 2014. UnityMol: Interactive scientific visualization for integrative biology. In: *2014 IEEE 4th Symposium* *on Large Data Analysis and Visualization (LDAV)*. 109–110. DOI: 10.1109/LDAV.2014.7013213.

Sehnal D, Deshpande M, Svobodová Vařeková R, Mir S, Berka K, Midlik A, Pravda L, Velankar S, Koča J. 2017. LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. *Nature Methods* in press.

Sehnal D, Rose AS, Koča J, Burley SK, Velankar S. 2018. Mol*: Towards a Common Library and Tools for Web Molecular Graphics. In: *Proceedings of the Workshop on Molecular Graphics and Visual Analysis of Molecular Data*. MolVA '18. Goslar Germany, Germany: Eurographics Association, 29–33.

Sekhavat YA, Zarei H. 2016. Enhancing the Sense of Immersion and Quality of Experience in Mobile Games Using Augmented Reality. *Journal of Computing and Security* 3:53–62.

Simkovic F, Ovchinnikov S, Baker D, Rigden DJ. 2017. Applications of contact predictions to structural biology. *IUCrJ* 4:291–300. DOI: 10.1107/S2052252517005115.

Smith JS, Isayev O, Roitberg AE. 2017. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science* 8:3192–3203. DOI: 10.1039/c6sc05720a.

Sommer B, Baaden M, Krone M, Woods A. 2018. From Virtual Reality to Immersive Analytics in Bioinformatics. *Journal of Integrative Bioinformatics* 15. DOI: 10.1515/jib-2018-0043.

Stocks MB, Hayward S, Laycock SD. 2009. Interacting with the biomolecular solvent accessible surface via a haptic feedback device. *BMC structural biology* 9:69. DOI: 10.1186/1472-6807-9-69.

Stocks MB, Laycock SD, Hayward S. 2011. Applying forces to elastic network models of large biomolecules using a haptic feedback device. *Journal of Computer-Aided Molecular Design* 25:203–211. DOI: 10.1007/s10822-010-9410-0.

Stone J. 2019. *VMD support for VR and Interactive MD*. https://www.ks.uiuc.edu/Research/vmd/allversions/interactive_MD.html: https://www.ks.uiuc.edu/Research/vmd/allversions/interactive_MD.html.

Stovgaard K, Andreetta C, Ferkinghoff-Borg J, Hamelryck T. 2010. Calculation of accurate small angle X-ray scattering curves from coarse-grained protein models. *BMC bioinformatics* 11:429. DOI: 10.1186/1471-2105-11-429.

Vega Garzón JC, Magrini ML, Galembeck E. 2017. Using augmented reality to teach and learn biochemistry. *Biochemistry and Molecular Biology Education: A Bimonthly Publication of the International Union of Biochemistry and Molecular Biology*. DOI: 10.1002/bmb.21063.

Wiebrands M, Malajczuk CJ, Woods AJ, Rohl AL, Mancera RL. 2018. Molecular Dynamics Visualization (MDV): Stereoscopic 3D Display of Biomolecular Structure and Interactions Using the Unity Game Engine. *Journal of Integrative Bioinformatics* 15. DOI: 10.1515/jib-2018-0010.

Wollacott AM, Merz KM. 2007. Haptic applications for molecular structure manipulation. *Journal of Molecular Graphics & Modelling* 25:801–805. DOI: 10.1016/j.jmgm.2006.07.005.

Wolle P, Müller MP, Rauh D. 2018. Augmented Reality in Scientific Publications—Taking the Visualization of 3D Structures to the Next Level. *ACS Chemical Biology* 13:496–499. DOI: 10.1021/acschembio.8b00153.

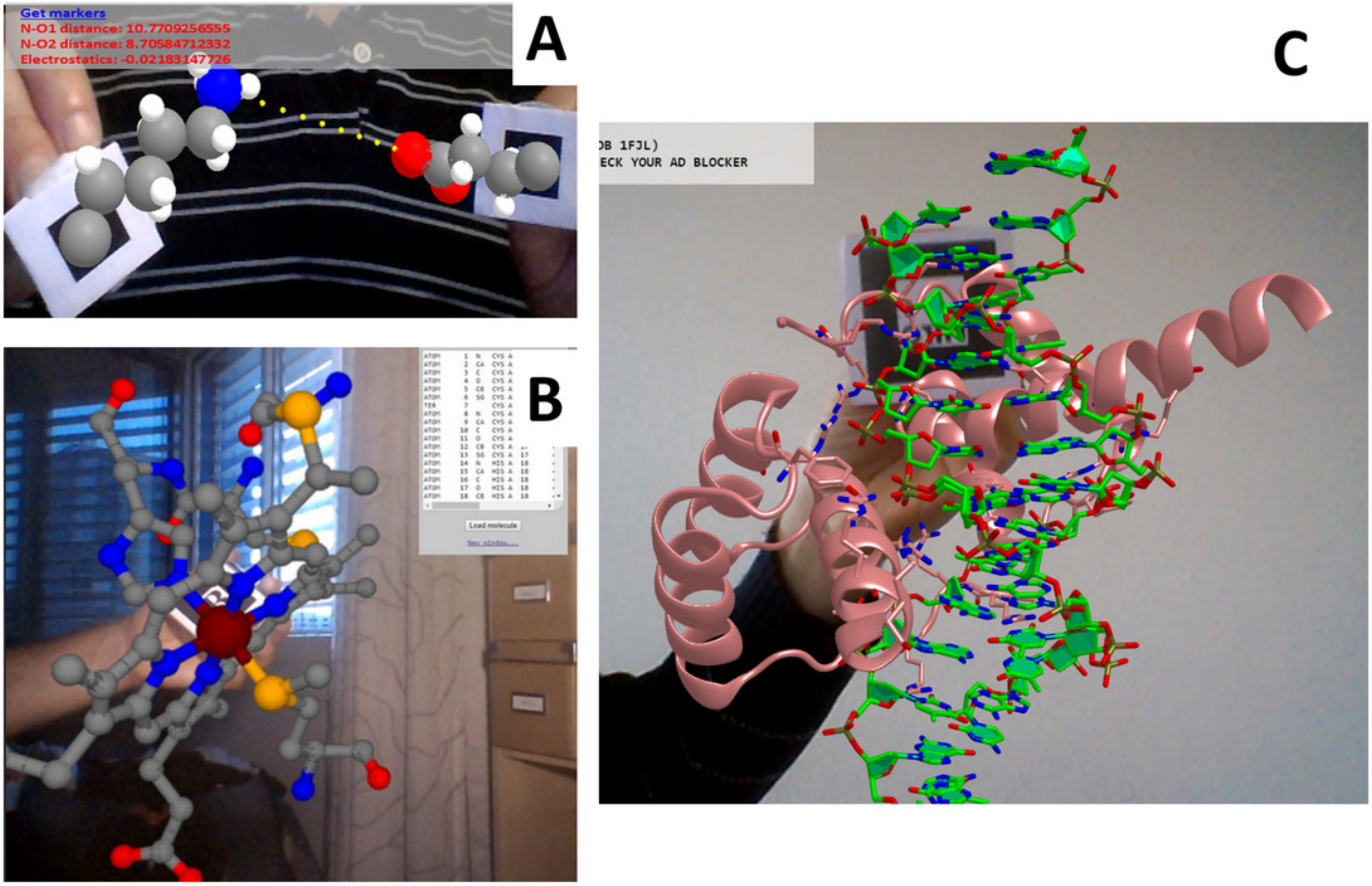
Y. Gaffary, B. Le Gouis, M. Marchal, F. Argelaguet, B. Arnaldi, A. Lécuyer. 2017. AR Feels “Softer” than VR: Haptic Perception of Stiffness in Augmented versus Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 23:2372–2377. DOI: 10.1109/TVCG.2017.2735078.

Yang S, Mei B, Yue X. 2018. Mobile Augmented Reality Assisted Chemical Education: Insights from Elements 4D. *Journal of Chemical Education* 95:1060–1062. DOI: 10.1021/acs.jchemed.8b00017.

Figure 1

Different implementations of WebGL for AR in web browsers.

(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 based in HTML-like tags from A-Frame. Accessible at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/smallmolclashdetection.html and video demo at https://www.youtube.com/watch?v=KOXDZ_cMjB0. (B) 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. Accessible at https://lucianoabriata.altervista.org/jsinscience/arjs/jsartoolkit5/pdbloader6.html with a video demo at https://www.youtube.com/watch?v=V5tJWREgGIg. (C) A Wavefront object exported from VMD with the OBJ/MTL renderer and loaded onto a marker using pure A-Frame code. The green molecule is a double-stranded segment of DNA, bound to a homeodomain shown as pink cartoons (PDB ID 1FJL), that id displayed spinning on the Hiro marker. Example accessible at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/obj-1fjl.html.



```
N-O1 distance: 10.7709256555
N-O2 distance: 8.70584712332
Electrostatics: -0.02183147726
```

```
ATOM      1  N   CYS A
ATOM      2  CA  CYS A
ATOM      3  C   CYS A
ATOM      4  O   CYS A
ATOM      5  CB  CYS A
ATOM      6  SG  CYS A
TER       7      CYS A
ATOM      8  N   CYS A
ATOM      9  CA  CYS A
ATOM     10  C   CYS A
ATOM     11  O   CYS A
ATOM     12  CB  CYS A
ATOM     13  SG  CYS A  17
ATOM     14  N   HIS A  18
ATOM     15  CA  HIS A  18
ATOM     16  C   HIS A  18
ATOM     17  O   HIS A  18
ATOM     18  CB  HIS A  18
```

```
DB 1FJL)
ECK YOUR AD BLOCKER
```

Figure 2

Tracking the real world with JavaScript libraries

(A) Kanji (left) and Hiro (right) markers that are built-in into AR.js (more sizes available in Figure S1). (B) Marker-based AR/VR facilitates concurrent view by multiple users. (C) WebGazer.js put to work to control JSmol, so that whatever region of the protein the user looks at comes to the front. I acknowledge Angel Herráez for help integrating both libraries. Example accessible at https://lucianoabriata.altervista.org/jsinscience/jsmolwebgazer/jsmolwebgazer.html .

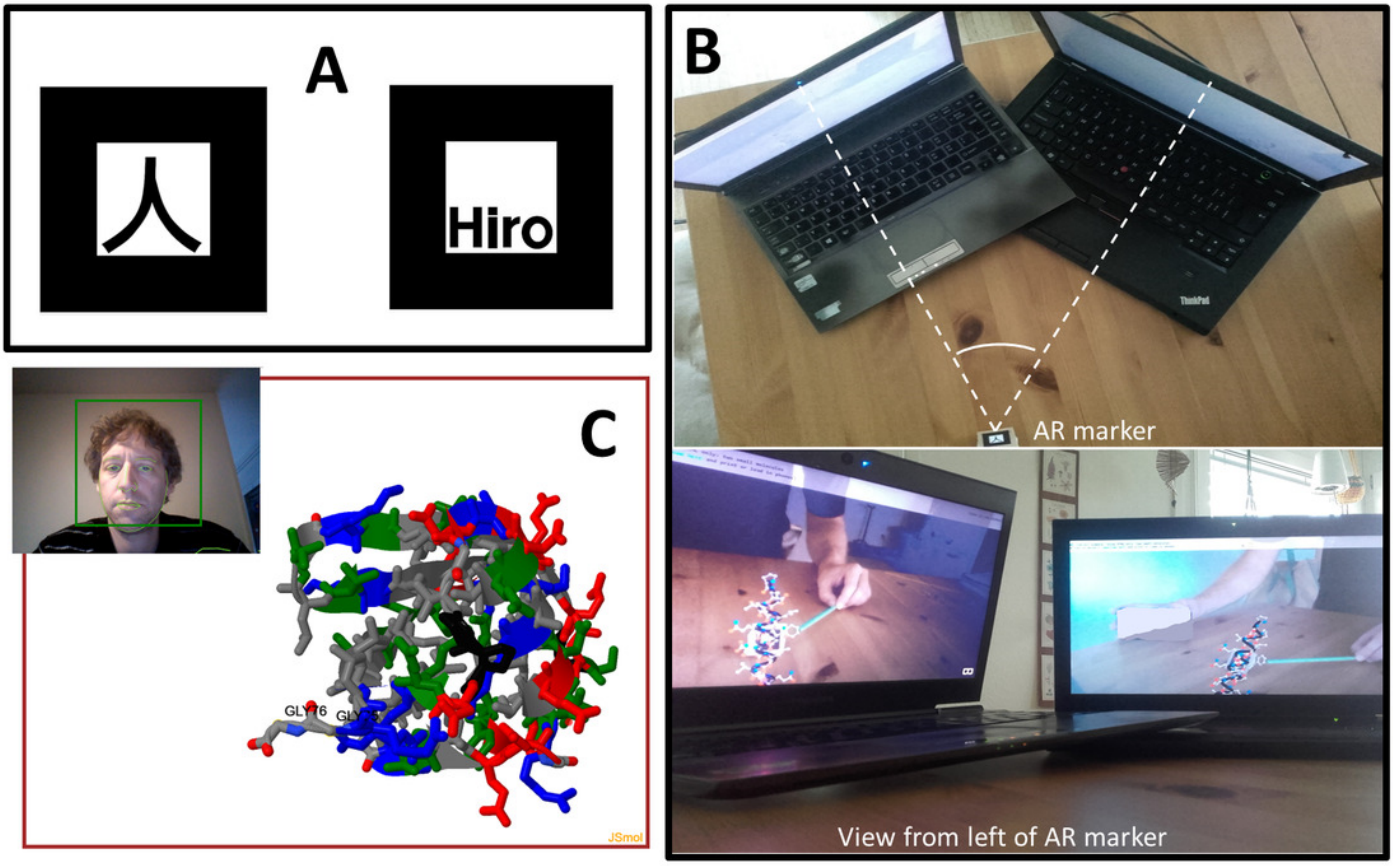


Figure 3

Interactivity in small molecules

(A) The product of a visual emulation of a Diels-Alder reaction, after the user moved the reagents close enough in space; this example further helps to visualize fused rings (as the diene reagent was a ring itself, leading to a double ring after reaction). Accessible at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/smallmoldielsalder.html and video demo at https://www.youtube.com/watch?v=SxB0GUbxmNA . (B) 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. Accessible at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/metmyoglobinfe3pcshift.html and video show at https://www.youtube.com/watch?v=Q_T_Oae1sE0 .

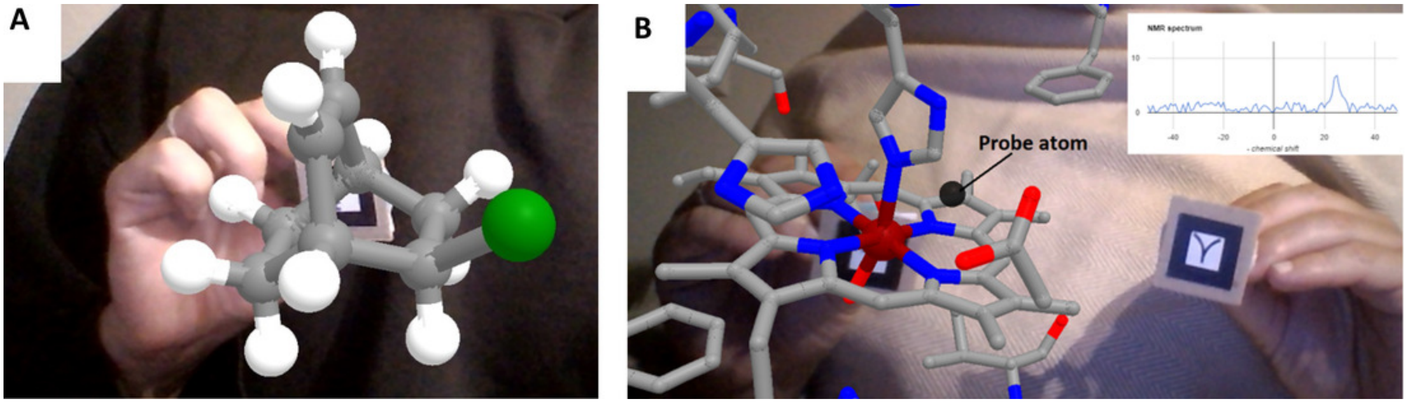


Figure 4

Visualization of biological macromolecules

(A) VMD-derived representation of PDB entry 1VYQ; this is essentially the same example shown by Berry and Board (*Biochem. Mol. Biol. Educ.* 2014) but in a ~~version that runs in~~ web browser~~s~~ with no ~~installs and no~~ plugins. Example accessible at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/obj-1vyq.html . (B) Representation of an amphipathic alpha helix built from primitives, viewable on the Hiro marker at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/ubiquitinandNESatomistic.html

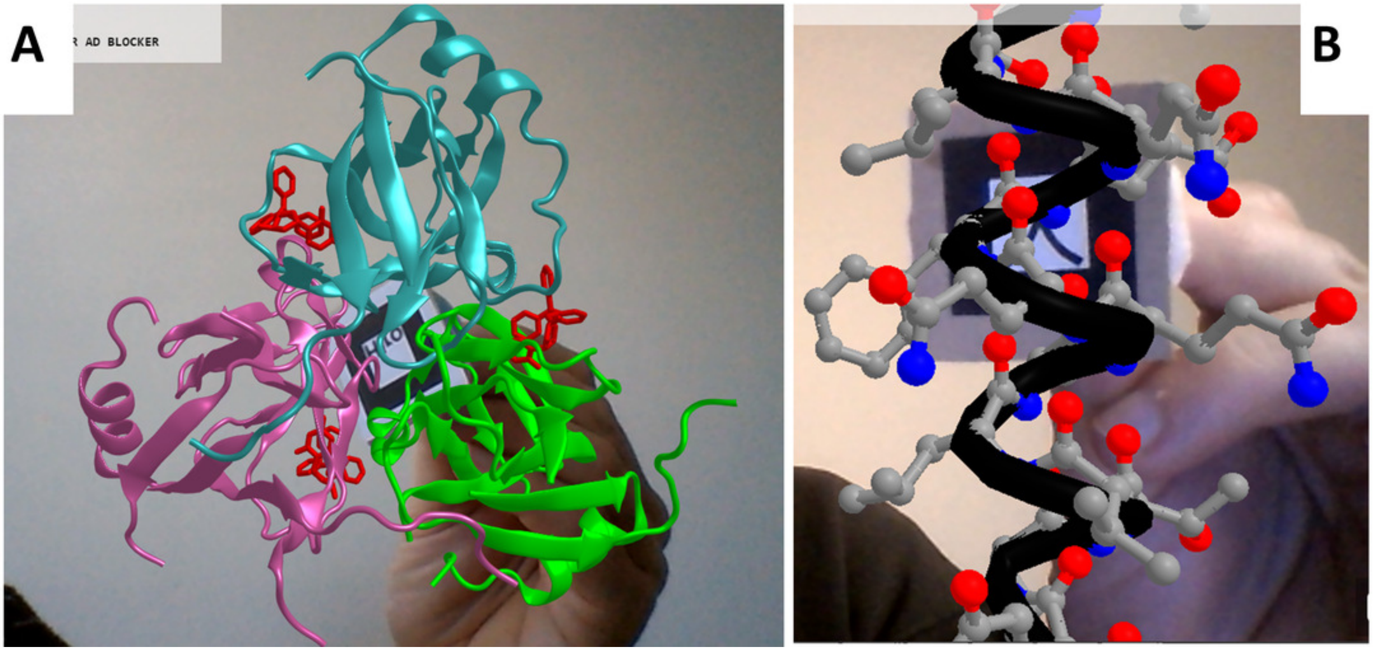


Figure 5

Interactive applications on biological macromolecules

Ubiquitin (red trace) and ubiquitin-interacting motif (blue) driven in 3D with two AR markers, as the web app computes the predicted SAXS profile and displays it overlaid on top of an experimental (simulated with noise) profile, together with a metric of the fit quality. This example can be tested at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/ubiquitinuimffvoicesaxs.html and a video demo is at https://www.youtube.com/watch?v=mLFXWTqhOd8 . (B and C) Interactive exploration of contacts predicted between two proteins (here from coevolution data) before (B) and after (C) 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. This example is available at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/coevol_1qop.html .

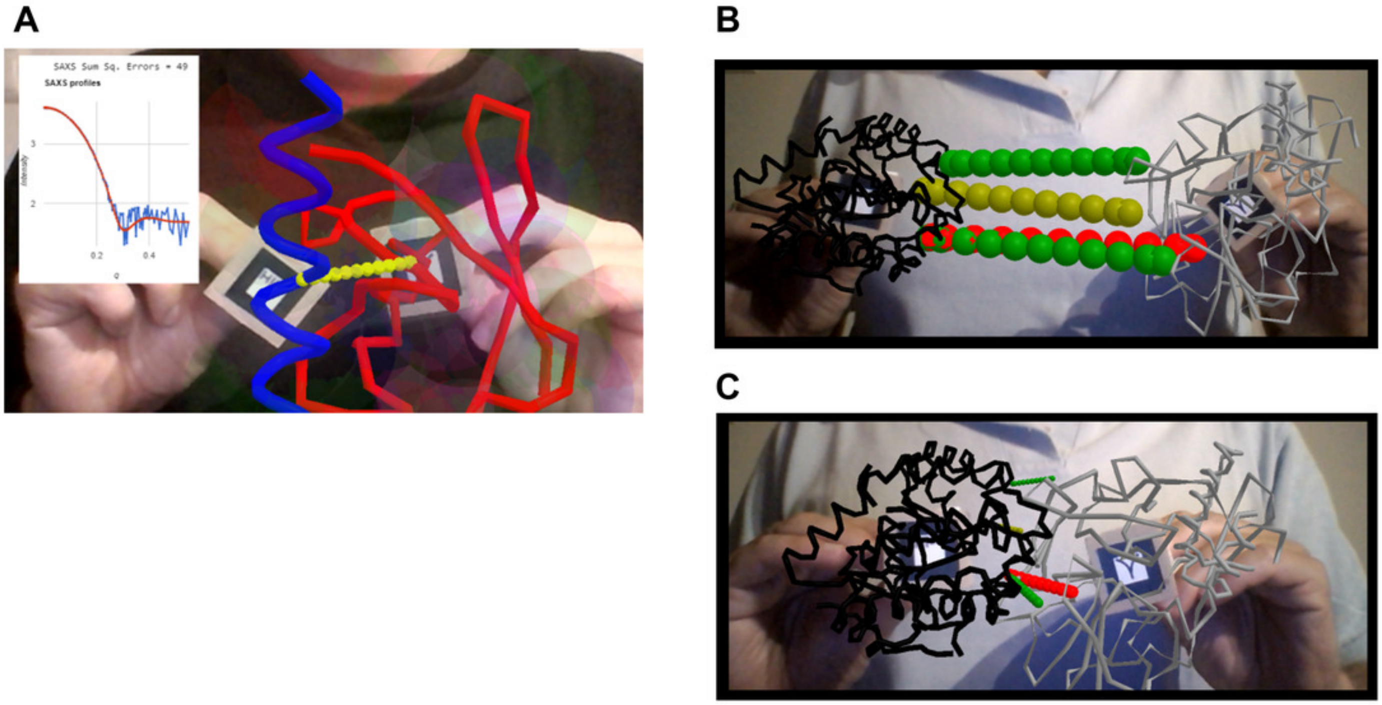

A
SAXS Sum Sq. Errors = 49
SAXS profiles
3
2
Intensity
0.2
0.4
q
B
C

Figure 6

Dynamics of highly disordered segments modeled with rigid-body mechanics

Ubiquitin and ubiquitin-interacting motif 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. This example is available at https://lucianoabriata.altervista.org/jsinscience/arjs/armodeling/ubiquitin-uim-cannon.html and a video demo at https://www.youtube.com/watch?v=G7oJO4guF1I .

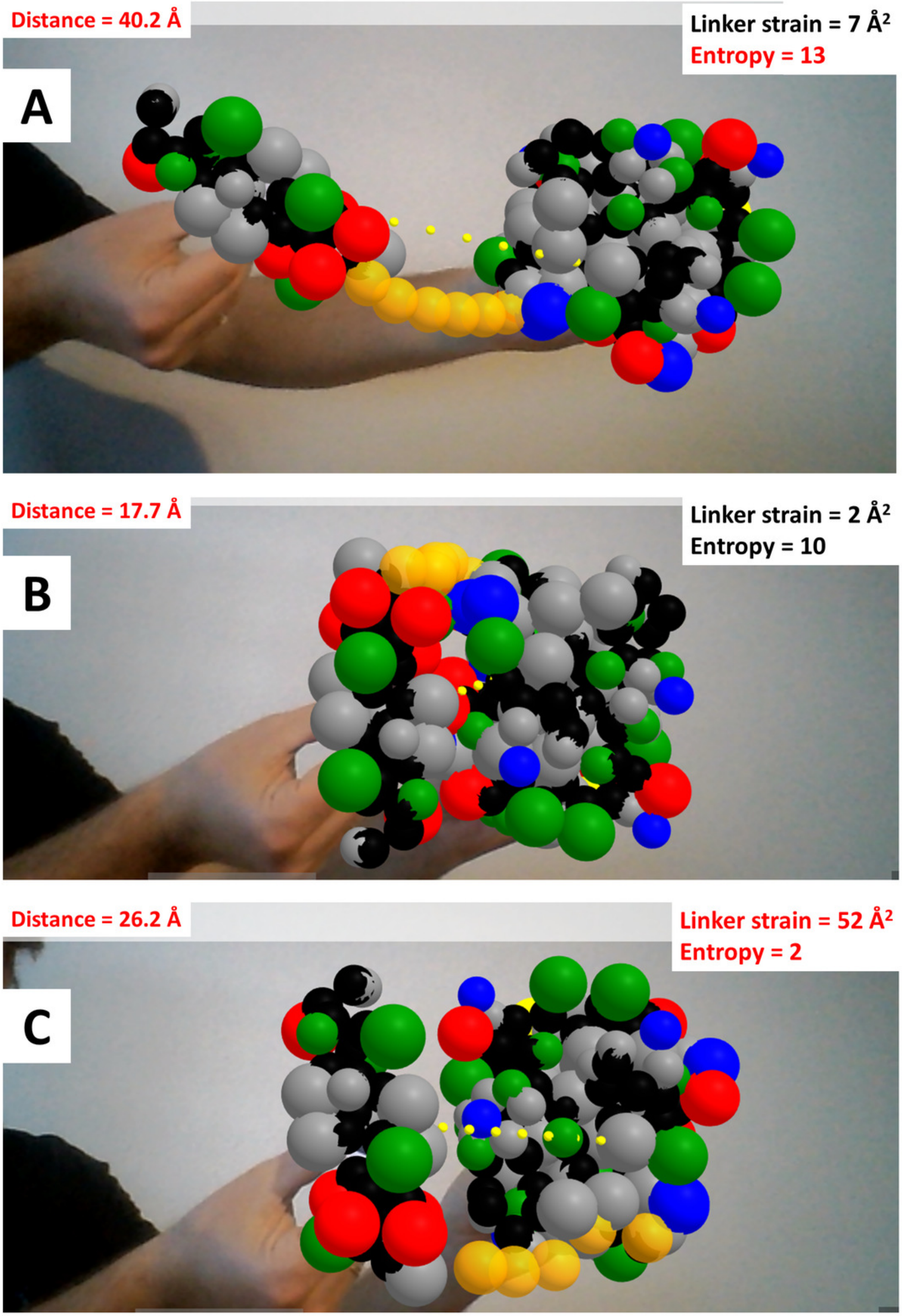

Distance = 40.2 Å
Linker strain = 7 Å²
Entropy = 13
A
Distance = 17.7 Å
Linker strain = 2 Å²
Entropy = 10
B
Distance = 26.2 Å
Linker strain = 52 Å²
Entropy = 2
C

Table 1(on next page)

Indications and troubleshooting for running and developing web-based AR contents

Table 1. Indications and troubleshooting for running and developing web-based AR contents

Hardware and Software

* Webcam- and Internet-enabled computer (desktop or laptop)
* Tested 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
* Ensure using https URLs (otherwise webcam will not activate)
* JavaScript and WebGL must be enabled in the browser (they are by default)
* Currently limited and heterogeneous support in tablets and smartphones, not recommended

Augmented reality markers

* Print on standard printer
* Avoid glossy papers, opaque paper is best
* Easier to handle if glued on solid surfaces (avoid wrinkles)
* Cubic marker glued on solid rubber cube cut to appropriate side

Additional troubleshooting

* Enable webcam when prompted
* Check that ad blockers do not block the webcam and other content
* Pages containing big Wavefront files may take time to load (half to a few minutes)
* Firewalls in your computer or surrounding network may prevent webcam activation

Development

* Need https server (free ones exist)
* Can also develop on localhost
* HTML-only using A-Frame and AR.js, placing atoms and bonds based on atomic coordinates or displaying premade objects. Or direct use of Three.js primitives
* Export premade objects in Wavefront format from popular molecular graphics programs like VMD
* Interactivity: use JavaScript on HTML identifiers of the objects that compose the scene