ProteinScanAR - An augmented reality web application for high school education in biomolecular life sciences

Stefan Nickels*, Hienke Sminia†, Sabine C. Mueller*, Bas Kools‡,
Anna Kathariona Dehof§, Hans-Peter Lenhof§ and Andreas Hildebrandt¶
**Intel Visual Computing Institute Saarland University, Saarbrücken, Germany*
Email: {nickels,mueller}@intel-vci.uni-saarland.de
†Netherlands Bioinformatics Centre, Nijmegen, Netherlands
Email: hienke.sminia@nbic.nl
‡Local Intelligence, Berlin, Germany
Email: hienke.sminia@nbic.nl
§Center for Bioinformatics, Saarland University, Saarbrücken, Germany
Email: {anna.dehof, len}@bioinf.uni-sb.de
¶Johannes-Gutenberg University, Mainz, Germany
Email: Andreas.Hildebrandt@uni-mainz.de

***Abstract*—Understanding protein structures is a crucial step in creating molecular insight for researchers as well as students and pupils. The enormous scaling gap between an atomic point of view and objects in daily life hampers developing an intuitive relation between them. Especially for high school students, it can be difficult to understand the spatial relations of a protein structure. Due to lack of direct imaging techniques, molecules can only be explored by studying abstract molecular models. Here, the use of Augmented reality (AR) techniques has proven to strongly improve structural perception. In this work we present ProteinScanAR, an augmented reality framework for biomolecular education that allows connecting virtual and real worlds intuitively, and thus enables focusing on the scientific or educational content. Special attention was taken to guarantee implementational and technical requirements as general and simple as possible to alleviate application in non-expert computer settings. The ProteinScanAR framework is freely available under the GNU Public License (GPL).**

***Keywords*-HTML5; 3D Internet; BALLView; Education; Augmented Reality; XML3D; Flash**

I. Background

Proteins build the essential molecular machinery in all living organisms such as bacteria, plants, fungi and animals including humans. Researchers examine these molecules in order to understand how nature operates and how this knowledge can be used.

Drug molecules, for example, are typically designed to inhibit protein molecules involved in diseases. Also, insights into the interaction network of proteins in the human body can help to reveal dependencies between food and human health.

For both applications, the three-dimensional structure of proteins plays a key role as subsumed in the famous quote by Francis Crick: “If you want to understand function, study structure” [1]. Only by looking at this structure are researchers able to recognize a protein’s binding site, or specific domains that allow for molecular interaction.

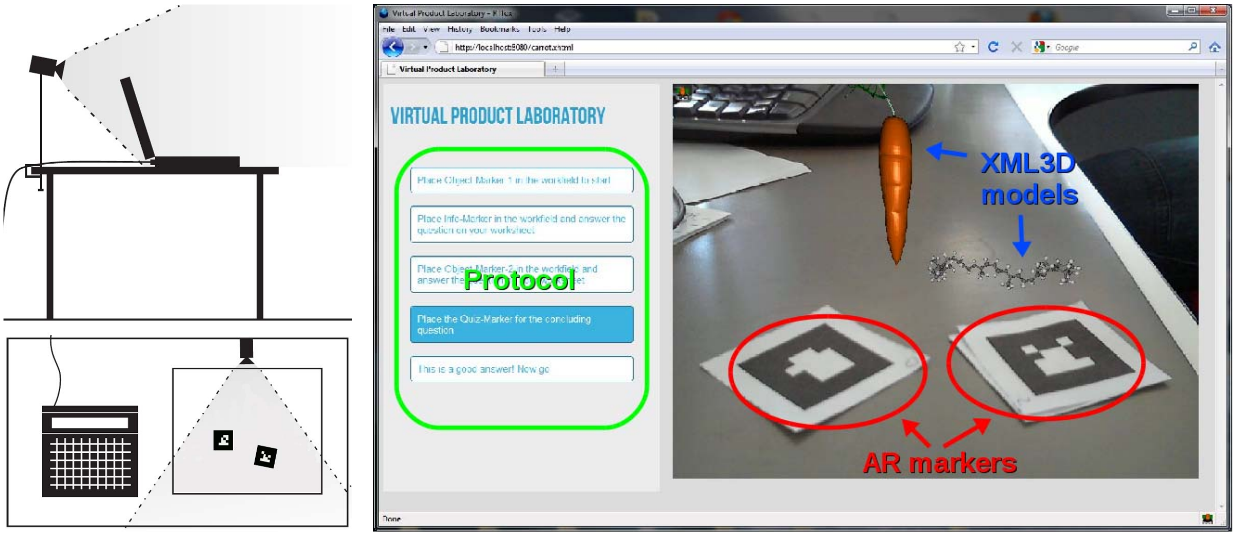
Thus, the geometric and physico-chemical understanding of a molecular structure is typically necessary for fully understanding its function. This necessity presents a problem: proteins and protein interactions are too small to be seen by far, even under advanced microscopes. Consequently, considerable research in Bioinformatics is devoted to developing better molecular visualization and molecular editing programs [2], [3], [4].

A common problem for researchers is to relate seemingly artificial molecular visualization to the normal 3D world. In a teaching context, this problem is even further aggravated, as pupils and students typically lack the required experience in working with biomolecular systems. Our ProteinScanAR project thus goes one step beyond former approaches and extends traditional molecular visualization by connecting both worlds: not only can a protein be visualized on a computer, but it can be used literally as a digital 3D scale model which users can examine in a virtual lab environment.

The tools set to interact with this augmented world is based on physical markers and objects detected by a webcam. These markers trigger actions and show three-dimensional molecular structures that are augmented to the recorded reality, allowing the user to conveniently dive into the atomic world of molecules without leaving their known reference frame.

Existing visualization programs for biomolecular scenes are mainly developed for scientists, which makes them hard to comprehend for the layman. In contrast, the application described in this work will be particularly designed with a teaching context in mind, and be mainly aimed at high school students. Experience has shown that for high school students, working with sophisticated and carefully designed

1550-6037/12 $26.00 © 2012 IEEE
DOI 10.1109/IV.2012.97


Virtual Product Laboratory - Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/carrot.xhtml
Virtual Product Laboratory
Protocol
XML3D models
AR markers
Done

VIRTUAL PRODUCT LABORATORY

Place Object Marker 1 in the workfield to start

Place Info-Marker in the workfield and answer the question on your worksheet

Place Object Marker-2 in the workfield and answer the [illegible] worksheet

Place the Quiz-Marker for the concluding question

This is a good answer! Now go

Figure 1. **Overview of the ProteinScanAR web application**: On the **left** you can see the typical desktop setup for ProteinScanAR (top and side view). On the **right** you see the ProteinScanAR website, with Protocol on the left and Flash module in the center.

AR applications is a very exciting work form which, pulls them into the subject, and can help in increasing not only motivation, but also understanding [5].

Educational research shows that emotional involvement (fun-factor) [6] and collaboration (pupils work in pairs) [7] improves learning. Thereby students can explore proteins on their own pace, enabling scaffolding between students of different levels of undestanding. The AR-activity gives teachers the possibility to do practical work within the subject of molecular biology. It is expected that this will make students more aware of the lessons learned, intensify their learning experience, and will improve their perception of depth in small molecules and cell processes.

The fact that our project focuses on pupils, and the requirement that the setup should be very easy to implement in practice in high school biology and chemistry classes, dictate some design choices:

- The application is web based to make it very easy and intuitive to use, since neither students, nor teachers, can be generally expected to have advanced technical expertise. Using a web browser also circumvents the problem that teachers are often not allowed to install software packages on school hardware. Smartphones were a reasonable option at first, but the availability of powerful models in each class cannot be guaranteed. In addition, the size constraints of a smartphone screen prevent the visualization of larger proteins with the desired level of detail. In contrast, tablets are a promising new device class, having sufficiently large screens and allowing for a flexible inspection of the markers. However, the requirement of having a camera integrated into the backside of the casing must be fulfilled.
- The content of the lessons provided by the application has to be linked to the school curriculum. Hence, it should be as simple as possible to set up new lesson scenarios to adapt to varying curricula.

A number of studies have shown that teaching with AR improves the spatial imagination of students [8], [9], which is crucial to understand molecular mechanisms in a living cell [10]. Not only high school students can benefit from this application: most molecular biologists and life scientists are not bioinformaticians as well, which makes it hard for many of them to use existing visualization software.

II. The ProteinScanAR Web Application

To provide an augmented reality web application that is suitable for biology high school education, we had to combine the expertise of teaching experts, product designers, and bioinformaticians alike. This unique combination lead to a number of challenging problems, that are discussed in the following. The result of this endeavour is our *ProteinScanAR* web application.

A. Module Overview

The user interface of the ProteinScanAR web application as depicted in Figure 1, links three basic aspects: content, visualization, and interaction.

The center of the website consists of a Flash container showing the webcam video.

On the left side, the page displays a protocol guiding the user through a lesson with the current step highlighted. A protocol step typically requests the user to place a certain AR marker into the view field of the camera to trigger the next action and to advance to the next step. As soon as the marker is recognized in the video stream, a certain object of interest in the lesson module will appear in the scene, typically by overlaying a 3D model onto the video stream relative to the position of the AR marker. This gives the impression that the model is hovering on top of it. In our current scenarios, the displayed objects are either food product models or molecular models which play a crucial role in the food product covered by the lesson. The course of an exemplary lesson (Carrot/betacarotene) can be found in Table I.

B. Core components

1) Augmented Reality in Flash: The term Augmented Reality refers to the scenario that a real-world environment, in our case a scene pointed at with a webcam, is augmented with 3D graphic models placed on top of detected marker patterns in the scene. Today, one of the popular system for the creation of such AR applications is the so called NyARToolkit [11] tracking library, which is in turn based on the original ARToolkit [12] library. By tracking the real world camera position and orientation relative to 2D physical markers in the scene in real time, these libraries allow drawing 3D graphics models perfectly overlaid on the real marker.

Implementing AR applications for the web was greatly simplified with the introduction of Tomohiko Koyama's FLARToolkit library [13] in 2009, which is a Flash/ ActionScript3 implementation of NyARToolkit. The great advantage of using Flash for such applications originates from a simple access to cameras connected to a user's computer in an operating system independent manner. Thus, harnessing inexpensive webcams for AR purposes is nearly trivial in Flash.

In contrast to a technically similar approach of Keil and Zoellner [14] with an AR web application using FLARToolkit and X3DOM [15], in our module we chose the FLARManager framework [16]. FLARManager is a lightweight framework that allows to easily build AR applications for Flash by supporting a variety of different Flash tracking libraries and Flash 3D frameworks. It supports the original FLARToolkit tracking library, which we also use in this module. One main advantage of FLARManager in contrast to using FLARToolkit directly is that the former allows for a flexible configuration of marker patterns without the need of touching a Flash IDE. The user only needs to add his preferred markers to a specific folder of the website and adapt an XML configuration file. This enables web designers or even the teachers themselves to modify the lessons without touching the implementation of the AR kernel, and without the requirement for a Flash compiler.

In our original design, we intended to use the visualization framework Papervision3D [17], which is embedded in FLARToolkit, to render the three-dimensional models. However, in Flash-based renderers the support of vertex coloring, required by most representation styles of molecules, such as solvent-excluded-surfaces (SES), turned out to be very challenging. Hence, we decided to use Flash only for the AR tracking capabilities of the FLARToolkit module, while using XML3D [18] for loading and rendering three-dimensional models in the browser.

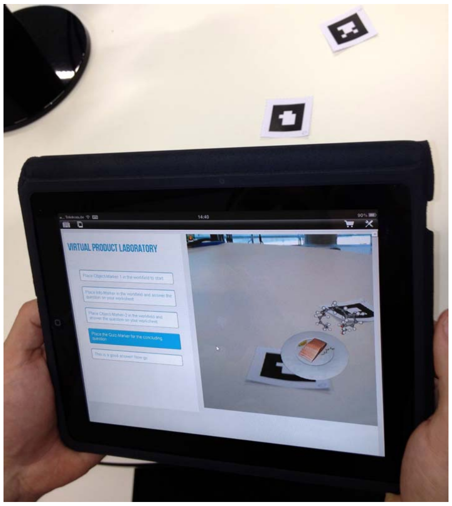

14:40
VIRTUAL PRODUCT LABORATORY
Place Object-Marker 1 in the worldfield to start
Place Info-Marker in the worldfield and answer the question on your worksheet
Place Object-Marker-2 in the worldfield and answer the question on your worksheet
Place the Quiz-Marker for the concluding question
This is a good answer! Now go

Figure 2. ProteinScanAR on a tablet device showing a lesson on Salmon and Omega-3 fatty acids

2) XML3D: XML3D, a recent approach for bringing interactive 3D graphics to the web, is an extension to HTML designed to describe 3D objects directly inside the source code of a website. It leverages existing web technologies such as HTML [19], Cascading Style Sheets (CSS) [20], the Document Object Model (DOM) [21] and JavaScript [22]. Hence, users are able to create and manipulate 3D content without additional web programming skills. XML3D is a purely declarative approach, where 3D content such as mesh models can directly be defined inside the DOM of a website using an XML syntax. This guarantees full support for DOM scripting and events for the manipulation of the 3D objects. As XML3D is an ongoing research project, support for XML3D is at the moment limited to native implementations of the Google Chrome and Mozilla Firefox browsers. However, a WebGL renderer written in JavaScript is offered by the XML3D project that can currently be used to enable the display of XML3D in every WebGL capable browser.

3) BALLView: All molecular models intended for display in the ProteinScanAR web application require XML3D mesh objects. In order to easily create XML3D models for molecules, we extended the molecular visualization tool BALLView[23], [3] to provide the export of molecular structures to the XML3D format.

BALLView is the graphical user interface to the Bio-

Protocol step text	Module action screen	Didactical intent
Place OBJECT-MARKER I in the work field to start.	A Carrot model appears, hovering above the marker.	Learn about the food product (carrot) of interest by rotation and zooming.
Place INFO-MARKER in the work field and answer the question on your worksheet.	A text box with an informational text about the carrot appears.	Learn key facts about the carrot, already pointing at the molecule of interest (beta-carotene).
Place OBJECT-MARKER II in the work field and answer the question on your worksheet.	The betacarotene molecule appears on the marker and can be freely rotated and zoomed.	Explore the structural features of betacarotene.
Place the Quiz-Marker for the concluding question.	HTML quiz appears-	Answer question about structural features of betacarotene. Next step is revealed on right answer.
This is a good answer.	Quiz Dialog disappears.	AR Lesson finished, A link to a follow-up website is revealed.

Table I
SEQUENCE OF THE "CARROT" LESSON

chemical Algorithms Library (BALL) [24], [25], and as such offers direct access to state-of-the-art molecular modeling techniques. The results of a number of modeling algorithms can then be directly visualized. A typical BALLView work flow, for instance, involves the download of molecular structures from established molecular data sources, such as the PDB or PubChem. Then, different representations (e.g. molecular surfaces, backbone, ribbons, or atom-based ball-and-stick models) can be added and combined to stress or simplify aspects and dependencies. Simulation functionality and energy evaluations allow estimations of fitness and probable survival of molecular complexes. All these individual steps are supported in BALLView with an intuitive interface. Hence, in creating a novel lesson module, the user merely has to decide on the molecular system he wants to display. He can then download the corresponding files from the respective data bases, prepare them (e.g., remove water, add missing atoms, and possibly minimize the structure), and test different visualization models. Once he is satisfied with the visual representation, he can export it from BALLView's export menu to an XHTML website embedding a XML3D element containing the scene. The corresponding XML3D mesh element in the exported XHTML website can be identified via the name that has previously been given to the corresponding molecular representation in BALLView.

C. Data Processing and Control using JavaScript

The task of interpreting the tracking information generated within the Flash AR tracking module and interpreting this information for triggering actions is realized using JavaScript. To this end, the tracker information first has to be extracted from the Flash module. The tracker module continuously generates positional and orientational information for the visible AR markers in the form of 4 coordinate transformation matrices. Furthermore each visible marker pattern can be identified by a `PatternID` and a unique `MarkerID`. In order to have this data available in the DOM of the website, we use the `External-Interface` API inside the Flash module to trigger a JavaScript function called `onFlashMarkerEvent` on every marker update. `onFlashMarkerEvent` is defined inside HTML and takes a position vector T, a rotation in an axis/angle representation R, a scaling vector S, as well as the `PatternID` and the `MarkerID`.

The whole logic of the module is then controlled from inside the `onFlashMarkerEvent` function. In reaction to the input `PatternID`, XML3D mesh objects with matching id's can be easily displayed by changing their `display` CSS style attribute and applying the transformation vectors to that object. Alternatively, arbitrary website content can be shown or manipulated as desired by the lesson designer. In practice, performing an expensive operation, such as transforming or rendering a molecular model, or sophisticated manipulation of the website's content, has an undesirable effect on the responsiveness of the application: the `ExternalInterface`-calls are blocking, and Flash is essentially single threaded. Thus, a video stream freezes during execution of the JavaScript call. We thus adopt a consumer-producer strategy, where the call of `onFlashMarkerEvent` does not directly trigger expensive JavaScript operations but instead deposits information in the DOM of the page. This information is then regularly evaluated by JavaScript-functions that have been installed using `setInterval`. One such function, e.g., evaluates a flag in the DOM to determine whether coordinates have changed and, if so, retrieves the new coordinates and triggers the transformation of the molecular model.

The comfort of only having to change a single JavaScript function and the need of only basic web programming skills in HTML and JavaScript for adapting a lesson make this web application suitable for teachers in high school education.

III. DISCUSSION

In the following, we discuss the overall performance of our web application ProteinScanAR especially with regard to usability, tracking performance and interaction between the core components. Furthermore, we present first impressions

Figure 3. Impressions of the teacher workshop and the initial user study.

that have been collected during first tests of the module in an educational environment.

A. Suitability, Usability, Configurability

The key features of our ProteinScanAR web application are the suitability for education accompanied by the configurability of the module. Since the typical users of this application are high school teachers of life science subjects with typically no or only limited knowledge in the field of classical application development, it is crucial that the configuration of the module is kept as simple as possible. Furthermore, building the application on standard and upcoming web technologies will make the application easily accessible, distributable, and usable even on performance limited school computers or mobile devices such as tablets. In addition, the application offers the possibility to create elaborate and demanding lessons modules for life sciences education, being able to use the full set of visualization features that BALLView offers. These findings are also confirmed by opinions collected at an initial user study among teachers.

Initial User Study: On March 9th, a prototype of the application has been tested with 16 high school teachers. In general, the teachers responded very positively. They found it really exciting to work with this new technology and were eager to test it in class. First reactions were: ’Very promising!’, ’This is an eye-opener to my students’ and ’Proteins are very hard to make tangible, and this can help a lot’.

Although teachers were enthusiastic, some were also critical: ’Before I can use this in my class the application should work faster and smoother’ and ’The application should be more easy to use’. This suggests that further tests are needed to point out what makes the settings and interface as convenient for teachers and students as possible. Due to the modular design of the application and the separation of programming logic from educational content, the ease of use can be easily improved as soon as additional feedback has been provided. In the following, we will turn to the more challenging aspect, the smoothness of the application.

B. Performance and component interaction

As the initial user study indicates, the application needs further improvement in performance. A first important step has already been made - while the user study was performed with a blocking implementation of compute intense JavaScript functions, we have since switched to the consumer -producer pattern described earlier. But still, several bottlenecks remain. While in our previous implementation, communication between Flash and JavaScript contributed to the overall run time significantly, the responsiveness of the application is now mostly governed by the tracking performance of the FLARManager. Testing the module on a typical ”school” hardware notebook (Intel Core2Duo, 2 GB RAM, integrated graphics) showed a FLARManager video performance of 4-5 fps, when tracking visible marker patterns. Interestingly, the rendering of the three-dimensional content itself has a much smaller effect on the application performance as experienced by the user, since rendering the structures does not cause the video stream to freeze and hence has a much less disturbing impact. The final overlay of the rendered structure on top of the video stream, however, again causes a small freeze as the display has to be synchronized between Flash and JavaScript.

IV. Conclusions and Outlook

In this work, we present ProteinScanAR, an AR based web application specificially aimed at high school education in the field of biomolecular life sciences. We have shown that ProteinScanAR allows simple creation of demanding and appealing lessons on molecular biology topics by use of AR. Learning content in the form of molecular structures can easily be prepared using the extensive molecular modeling and visualization program BALLView and subsequently be exported to XML3D for direct use in the application. The strict separation of programming logic and learning content keeps needed preparation effort low for teachers. This separation furthermore guarantees that new developments in the field of web technologies can easily be adopted into the application. Until recently, Flash has been the only reliable way for accessing a webcam through the browser, as native access to the camera using plain HTML, even with HTML5 features, was not possible. However camera access through JavaScript has now become available via the new WebRTC [26] framework. WebRTC, short for "Web Real-Time Communications", is a real-time media framework in the browser core providing JavaScript APIs for accessing and controlling network, audio, and video components. At the moment, WebRTC can be tested in experimental builds of the Google Chrome browser. Combined with already available JavaScript implementations of NyARToolkit this will finally allow a pure HTML5 implementation, which will further improve the simplicity and help to further improve the performance of ProteinScanAR.

Acknowledgments

We want to thank the students that contributed to the project: Kai Stenkamp (initial HTML implementation), Eric Stens (food product 3D models), Tomas van Nassau (input didactic scenario) and Niels Buddiger (CSS layout and follow-up website). This work was granted by the Netherlands innovative-programme SURFnet/Kennisnet, the Schwerpunkt "Rechnergestützte Forschungsmethoden in den Naturwissenschaften" of Johannes-Gutenberg University Mainz and the Intel Visual Computing Institute of Saarland University.

References

[1] F. Crick. *What Mad Pursuit: A Personal View of Scientific Discovery*. 1988.

[2] W.L. DeLano. PyMOL - a user-sponsored molecular visualization system on an open-source foundation. http://www.pymol.org.

[3] A. Moll, A. Hildebrandt, H.-P. Lenhof, and O. Kohlbacher. BALLView: a tool for research and education in molecular modeling. *Bioinformatics*, 22(3):365–366, 2006.

[4] E. Pettersen, T. Goddard, C. Huang, G. Couch, D. Greenblatt, E. Meng, and T. Ferrin. UCSF Chimera–a visualization system for exploratory research and analysis. *J Comput Chem*, 25(13):1605–1612, Oct 2004.

[5] J. H. Mathewson. Visual-spatial thinking: An aspect of science overlooked by educators. *Science Education*, pages 33–54, 1999.

[6] M. Immordino-Yang and A Damasio. Does active learning work? a review of the research. *Journal of Mind*, 1(1):3–10, 2007.

[7] P. Dillenbourg. *Collaborative Learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series*. Elsevier Science, Inc., New York, 1999.

[8] K. Lee. Augmented reality in education and training. *TechTrends*, 56(2):13–21, 2012.

[9] K.J. Schoenborn and T.R. Anderson. The importance of visual literacy in the education of biochemists. *Biochemistry and Molecular Biology Education*, pages 94–102, 2006.

[10] B. Dalgarno and Lee M.J.W. What are the learning affordances of 3-d virtual environments? *British Journal of Educational Technology*, pages 10–32, 2006.

[11] R. Iizuka. NyARToolkit. http://nyatla.jp/nyartoolkit/wp/, 2008.

[12] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, IWAR '99, pages 85–, Washington, DC, USA, 1999. IEEE Computer Society.

[13] T. Koyama. FLARToolkit. http://www.libspark.org/wiki/saqoosha/FLARToolKit/en, 2009.

[14] J. Keil and M. Zoellner. Flash + AR / X3DOM mashup. http://www.x3dom.org/?page_id=583, 2010.

[15] J. Behr, Y. Jung, J. Keil, T. Drevensek, M. Zoellner, P. Eschler, and D. Fellner. A scalable architecture for the HTML5/X3D integration model X3DOM. pages 185–194, 2010.

[16] E. Socolofsky. FLARManager: Augmented reality in flash. http://words.transmote.com/wp/flarmanager/, 2011.

[17] Papervision3D, open source realtime 3d engine for flash. http://code.google.com/p/papervision3d/, 2007.

[18] K. Sons, F. Klein, D. Rubinstein, S. Byelozyorov, and P. Slusallek. XML3D - interactive 3d graphics for the web. pages 175–184, 2010.

[19] W3C. HTML5 specification. http://dev.w3.org/html5/spec/Overview.html, 2010.

[20] W3C. Cascading style sheets level2 revision 1 specification. http://www.w3.org/TR/CSS2, 2011.

[21] W3C. Document object model (dom). http://www.w3.org/DOM, 2009.

[22] ECMA. ECMAScript language specification. http://www.ecma-international.org/publications/standards/Ecma-262.htm, 2011.

[23] A. Moll, A. Hildebrandt, H.-P. Lenhof, and O. Kohlbacher. BALLView: an object-oriented molecular visualization and modeling framework. *Journal of Computer-Aided Molecular Design*, 19(11):791–800, November 2005.

[24] O. Kohlbacher and H.-P. Lenhof. Ball–rapid software prototyping in computational molecular biology. biochemicals algorithms library. *Bioinformatics*, 16(9):815–824, Sep 2000.

[25] A. Hildebrandt, A.K. Dehof, A. Rurainski, A. Bertsch, M. Schumann, N.C. Toussaint, A. Moll, D. Stockel, S. Nickels, S.C. Mueller, H.-P. Lenhof, and O. Kohlbacher. BALL - biochemical algorithms library 1.3. *BMC Bioinformatics*, page 531, 2010.

[26] WebRTC. http://www.webrtc.org, 2011.