

## 3D-Molekülvisualisierung im Internet Schwerpunkt Java-Applets

Andreas Bohne-Lang<sup>1</sup>, Elke Lang<sup>2</sup>

<sup>1</sup>Zentrale Spektroskopie, Deutsches Krebsforschungszentrum Heidelberg

<sup>2</sup>Fachbereich IuW, FH Darmstadt

► Die 3D-Visualisierung von Molekülen mittels Standardbrowsern wie Internet Explorer oder Mozilla ist ohne zusätzliche Software nicht möglich. Dieser Artikel gibt einen kleinen Überblick über die gängigen Techniken und legt dabei den Schwerpunkt auf Java-Applets.

### Einleitung

Unter der „Visualisierung von Molekülen“ versteht man viele verschiedene Formen, die sich zum einen in der Art der Darstellung, wie zum Beispiel zweidimensional in Form einer Strukturgrafik (Strichformel) oder dreidimensional in Form eines Kalottenmodells, und zum anderen im Inhalt der Darstellung unterscheiden. Bei der inhaltlichen Visualisierung kann man zum Beispiel physikochemische Eigenschaften wie Ladungsverteilung oder Lipophilie im Molekül hervorheben oder ein Protein nicht auf atomarer Ebene, sondern anhand seiner Tertiärstruktur einfärben. Allen Ansätzen der Visualisierung gemein ist, dass sie bestimmte Aspekte der molekularen Struktur hervorheben und somit zu deren Verständnis beitragen.

### Methoden und Techniken

Die aktuellen Webbrowser wie Mozilla, Internet Explorer, etc. beschränken sich bei ihrer Darstellung von Inhalten auf text- und

layoutbasierte Komponenten. Für alle weiteren Darstellungsmodalitäten benötigt man zusätzliche Software. Allgemein kann man drei grundlegende Techniken unterscheiden, wie diese zusätzliche Software mit dem Browser verbunden wird.

- Zum einen gibt es die Hilfsprogramme (helper applications), welche als externes Programm vom Webbrowser gestartet werden, vom Browser einen Pfad auf eine Datei übergeben bekommen, die bereitgestellte Datei laden und dann eigenständig, parallel zum Webbrowser laufen.
- Zum anderen gibt es zwei weitere Formen, bei denen die Software in den Browser integriert wird: Das Plug-in und das Java-Applet (davon abgesehen, dass die Virtual Machine, welche zum Ausführen des Java-Byte-Codes benötigt wird, in der derzeitigen Version als Plug-in in den Webbrowser geladen wird). Da dieses nicht immer so war, soll es an dieser Stelle getrennt betrachtet werden.

Die grundlegenden Eigenschaften der drei Verfahren sind in *Tabelle 1* aufgelistet. Als Beispiel soll die 3D-Visualisierung eines Moleküls stehen.

Java-Applets zeichnen sich dadurch aus, dass nach einmaliger Installation der virtuellen Java-Maschine auf dem Rechner jedes Java-Applet ausgeführt werden kann. Lediglich die Java-Erweiterung Java3D bedarf einer weiten Installation. Bindet man folglich ein normales (ohne Java3D) Java-Applet in eine Webseite ein, so kann der Benutzer auch ohne zuvor installierten Molekülbetrachter die 3D-Struktur des Moleküls betrachten.

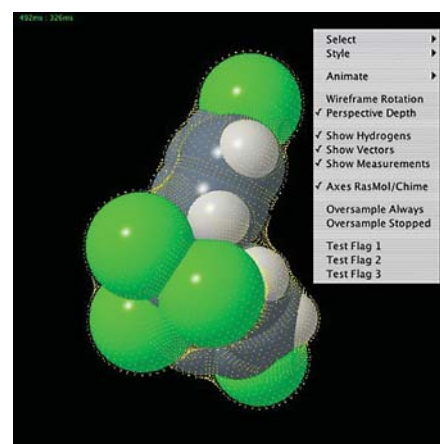


Abb. 1: Jmol[W13]

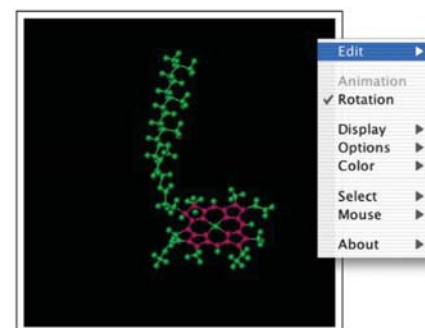


Abb. 2: Chemis3D[W12]

Tab. 1: Verschiedene Softwaretechniken der Molekülvisualisierung mit Hilfe von Browsern

	Java-Applet	Plug-in	Ext. Programm
Installation durch den Benutzer	Nein	Ja	Ja
Einfachheit der Installation	–	einfach	mittel
Betriebssystemabhängigkeit	gering	hoch	hoch
Browserabhängigkeit	gering	hoch	keine
Interaktion mit anderen Software-Komponenten	Ja	Ja	Nein
Geschwindigkeit	langsam-mittel	hoch	hoch
Übertragene Daten	Applet-Code und Moleküldatei	nur Moleküldatei	nur Moleküldatei
Parameter	mehrere	mehrere	einer
Layout	Browserfenster	Browserfenster	Extern
Möglichkeiten der Benutzerinteraktion	Ja	Ja	Ja
Typische Vertreter	WebMol[W14], Jmol[W13], Chemis3d[W12],	Chime[W3], canDo[W28]	RasMol[W1], PyMol[W2], Swiss-PDB Viewer[W4]

Es gibt mehrere in Java programmierte Molekülvisualisierer. Im Folgenden sollen die wichtigsten vorgestellt werden. Zuvor noch ein paar Worte zur Technik: Leider ist die Ausführungsgeschwindigkeit von Applets immer noch langsamer als die von Plug-ins wie Chime oder von externen Programmen. Auch wenn Java-Applets im Prinzip systemunabhängig sind, da hier ein Bytecode interpretiert wird, so kann es doch hin und wieder bei älteren Anwendungen zu Problemen kommen, da sich im Laufe der Entwicklung Kleinigkeiten in der Software-schnittstelle geändert haben. Als Beispiel benötigt der PDBViewer die Java-Run-time-Umgebung 1.4.2, welche noch nicht auf allen Rechnern installiert ist.

Manche der Java-Applets sind speziell für die Darstellung kleiner Moleküle geeignet, wohingegen andere speziell zum Anzeigen von Proteinstrukturen entwickelt worden sind. Bei den kleineren Molekülen zeichnen sich vor allem die Applets Jmol[W13] (Abb. 1), Chemis3D[W12] (Abb. 2), JaMM[W18], der Java 3d Molecule Viewer[W19] und das

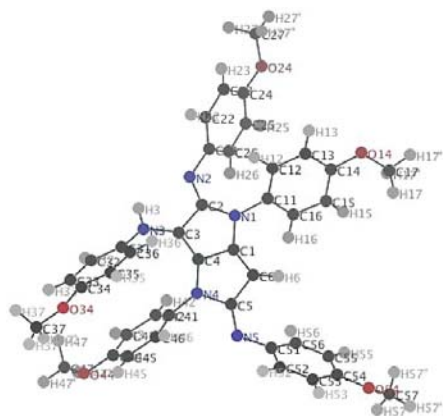


Abb. 3: 3d Java Molecule Viewer [W15]

xyzApp [W20] (von Sun) aus. Ein durch sein sehr schlichtes Layout bestechendes Applet ist der ‚3d Java Molecule Viewer‘ von Eric Harlow [W15] (Abb. 3). Hier werden nur die Atome, Atombindungen und Atomlabels angezeigt.

Jmol ist derzeit die am meisten vorangetriebene Entwicklung auf diesem Gebiet. Es werden mehrere Molekülformate zum Laden unterstützt und im klassischen Balls-and-Sticks-Modus angezeigt. Der Benutzer kann getrennt den Darstellungsmodus für Bindungen und Atome wählen. Ferne können die häufigsten Atomarten wie Kohlenstoff, Sauerstoff, Stickstoff, Schwefel und Wasserstoff separat markiert werden. Als Besonderheit ist hervorzuheben, dass Jmol in der Lage ist, Archivdateien abzuspielen. Ferner werden Chime-Skripte verarbeitet, und somit bietet Jmol sich aufgrund seiner Betriebssystemunabhängigkeit als echte Alternative zu Chime an. Es gibt auch ein Perl-Skript, mit dem man Chime-Skripte in ein Jmol-komformes Format umwandeln kann. Vieles von der Funktionalität ist noch nicht in das Menü von Jmol integriert und muss via Skriptbefehl aktiviert werden.

Chemis3D kann man als Schweizer Taschenmesser unter den Molekülviewer-Applets bezeichnen. Die Menüstruktur ist stark an die von Rasmol angelehnt, und man kann sowohl kleine Moleküle als auch Proteine darstellen. Als Darstellungsmodi stehen neben Backbone für Proteine die klassischen Darstellungsarten wie Wireframe, Balls and Sticks zur Verfügung. Die Einfärbung des Moleküls kann nach folgenden Kriterien vorgenommen werden: CPK, Residuen, Sharply, Chain, Temperatur und Struktur.

Ein klassischer Vertreter ist das Applet WebMol [W14] (Abb. 4) von Dirk Walter [Wa97], welches sowohl für kleine Moleküle als auch für Proteine geeignet ist. WebMol zeichnet sich auch dadurch aus, dass es beim

Testen wirklich bisher auf jedem Browser lief. Die besonderen Fähigkeiten liegen zum einen im Stereo-View-Modus, mit dem es möglich ist, einen plastischen Eindruck zu bekommen, und zum anderen kann WebMol eine Oberfläche durch eine Punktwolke andeuten.

MolVie [W17] (Abb. 5) ist für die Darstellung von Proteinen entwickelt worden und bietet dem Benutzer die Möglichkeit, den Aminosäuresequenzcode zu betrachten. Wenn man das Applet als ‚vertrauenswürdig‘ einstuft, kann man auch aus dem Browser heraus Dateien und Bilder speichern.

Die anderen Applets, wie das xyzApplet von Sun, der Java 3d Molecule Viewer von Anton Zamov oder JaMM, sind hier nur der Vollständigkeit halber aufgeführt.

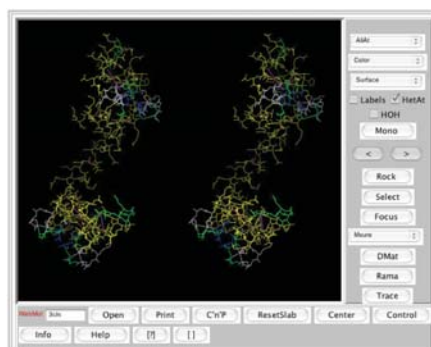


Abb. 4: WebMol-Applet im Stereo-Modus [W14]

### Java3D

Neu und leider nicht mehr ganz so kompatibel wie Java ist die Java-Erweiterung, Java3D [W22], welche derzeit von Sun nur für Windows und Solaris zum Herunterladen bereitgestellt wird. Ein Linux-Port existiert, und an einer Version für MacOSX wird ge-

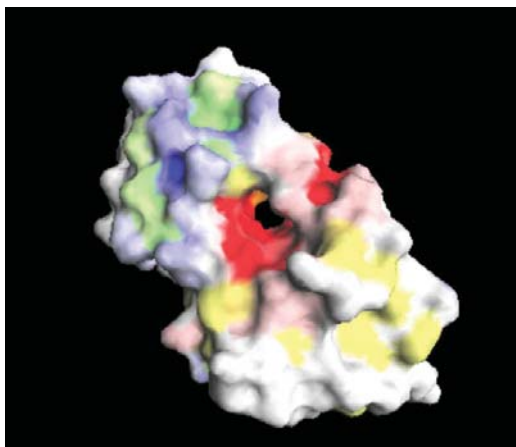


Abb. 6: PDBjViewer [W25]

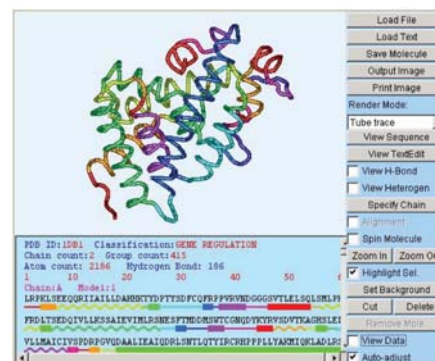


Abb. 5: MolVie [W17]

arbeitet. Java3D setzt entweder auf den 3D-Schnittstellen DirectX oder OpenGL auf und macht nur Sinn bei einer Grafikkarte mit 3D-Beschleuniger. Momentan kann man die 3D-Molekülvisualisierer, die auf Java3D aufsetzen, als experimentell bezeichnen. Mit Java3D kann man bei der Entwicklung von Molekülbetrachtern auf 3D-Primitive, welche von der 3D-Schnittstelle wie OpenGL bereitgestellt werden, zurückgreifen und die Darstellung einfach dem Treiber überlassen. Somit ist auch die Erzeugung von komplexen Objekten oder Oberflächen und das Auflegen von Texturen keine Hexerei mehr. Dementsprechend brillant ist die Darstellung der Moleküle (siehe Abb. 6 und 7).

### Plug-ins

Bisher ist Chime das am häufigsten nativ eingesetzte Plug-in. Leider ist Chime (Abb. 8) stark browserabhängig und nur nativ für Windows entwickelt. Aber aufgrund seiner Mächtigkeit hat es starke Verbreitung gefunden. Chime ging ursprünglich aus dem RasMol-Code hervor und wird frei (gegen Registrierung) von der Firma MDL vertrieben. Für Linux gibt es das Cross-Over-Plug-

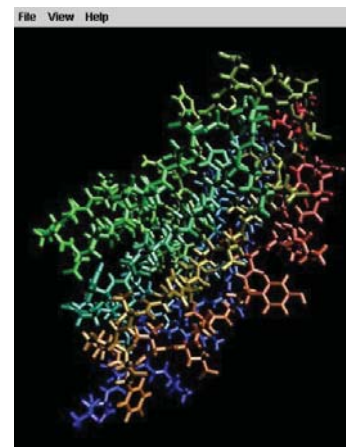


Abb. 7: JMV [W23] und Java3D [W22]

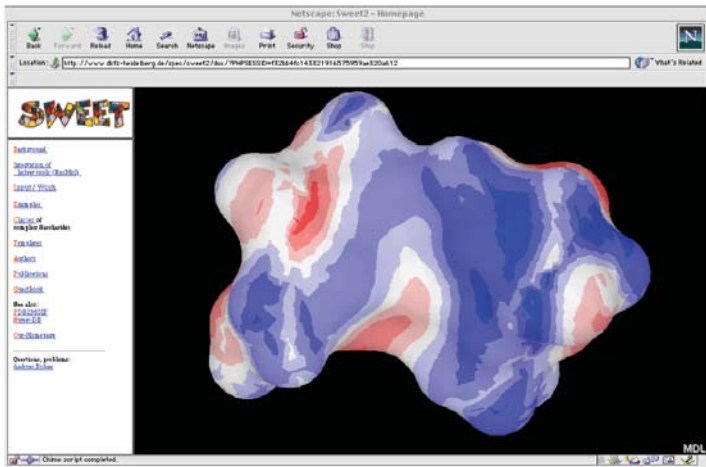


Abb. 8: Darstellung der Oberfläche eines Zuckers mittels CHIME [W3]

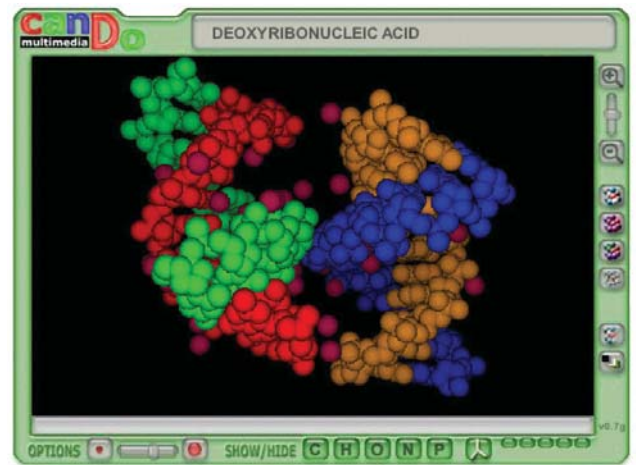


Abb. 10: Molekül via CanDo / Shockwave [W28]

in<sup>1</sup>, mit dem es möglich ist, Plug-ins, die für Windows entwickelt wurden, unter Linux zu benutzen.

Als eher exotisch kann man die Visualisierung via Autoren- oder Multimedia-Plug-ins bezeichnen. Dabei gibt es zwei Programme, derer man sich bedienen kann. Zum einen Flash, das von vielen anderen Webseiten bekannt ist und momentan das Standard-Multimedia-Plug-in ist. Hier gibt es derzeit zwei Umsetzungen. Zum einen den CML-Reader von [W26] und das eBooking Tool von Vera Fleischer [W27] (Abb. 9). Beide Programme kann man als rudimentär bezeichnen daher seien sie nur der Vollständigkeit halber erwähnt. Sehr viel interessanter wirkt da schon das canDo-Programm (Abb. 10) einer irischen Firma, welche auf das Shockwave-Plug-in aufsetzt. Wie auch beim Chime-Plug-in ist die Betriebssystemabhängigkeit hoch. canDo stellt dem



Abb. 9: Molekül via Flash [W27]

Benutzer eine Reihe von Interaktionsmöglichkeiten zur Verfügung.

### Zusammenfassung und Diskussion

Je nach System und Anwendungsproblem muss der Benutzer konkret entscheiden, welche Variante für ihn am sinnvollsten ist. Es lässt sich leider keine Empfehlung für Entwickler von Webseiten, welche 3D-Moleküle auf ihren Seiten anbieten, abgeben. Da Java-Applets betriebssystemunabhängig laufen, sollte man diesen den Vorzug gegenüber reinen Plug-ins wie Chime geben, wenn es um die Darstellung von Molekülen in Webseiten geht. Bei den Java-applets ist Jmol und Chemis3D am universellsten einsetzbar. Jmol besticht durch die Nähe zu Chime. Benötigt man aber besondere Funktionalitäten, so kann man andere Applets anbieten. Ob anstatt oder zusätzlich, bleibt dem speziellen Einsatzbereich überlassen. Beim Einsatz von Applets, welche auf die Java-Erweiterung 3D aufsetzen, sollte sichergestellt werden, dass der Benutzer eine Alternative nutzen kann, welche parallel angeboten werden sollte.

### Literatur

- [Bo00] **Bohne A., Lang E., von der Lieth C.W.** (2000): Molecular visualization on the web. *Drugs of the Future* 25: 489–500
- [Ga96] **Gasteiger J., Sadowski J., Schuur J., et al.** (1996): Chemical information in 3D-space. *J. Chem. Inf. Comput. Sci.* 36: 1000–8
- [Ih97] **Ihlenfeld W.** (1997): Virtual reality in chemistry. *J. Mol. Model.* 3: 386–402
- [IE98] **Ihlenfeld W., Engel K.** (1998): Visualizing chemical data in the internet – Data-driven and interactive graphics. *Comput. Graphics* 22: 703–14

[Ki03] **Kinoshita K., Nakamura H.** (2003): eF-site and PDBViewer: Database and Viewer for Protein Functional Sites. *Bioinformatics*, in press

[VB95] **Vollhardt H., Brickmann J.** (1995): 3D molecular graphics on the world wide web. *Pac. Symp. Biocomput.*: 663–73

[Su95] **Sun H., Li M., Xu Y.** (2003): MOLVIE: an interactive visualization environment for molecular structures. *Comput Methods Programs Biomed.* 71: 85–90.

[Ta03] **Tate J.** (2003): Molecular Visualization; *Methods Biochem Anal.*: 135–158

[Wa97] **Walther D.**: WebMol – A Java-based PDB viewer. *Trends Biochem. Sci.* 2: 274–5

Weiterführende URLs [W1–W28] unter:  
[www.dkfz.de/spec/suppl\\_mat/bs04](http://www.dkfz.de/spec/suppl_mat/bs04)

### Korrespondenzadresse:

**Dr. Dipl.-Inform.**  
**Andreas Bohne-Lang**  
Zentrale Spektroskopie / B090  
Deutsches Krebsforschungszentrum (DKFZ)  
Im Neuenheimer Feld 280  
D-69120 Heidelberg  
[a.bohne@dkfz-heidelberg.de](mailto:a.bohne@dkfz-heidelberg.de)

<sup>1</sup> <http://www.codeweavers.com/products/crossover/>