Article

molVisual3D: A standalone executable software for 3D visualization of molecules

WenJun Zhang
School of Life Sciences, Sun Yat-sen University, Guangzhou, China
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 20 June 2024; *Accepted 5 July 2023*; *Published online 30 July 2023*; *Published 1 June 2024*

Abstract
In this article, a standalone executable software molVisual3D, for 3D visualization of molecules was developed. It uses the XYZ file of a molecule to generate its 3D graphics. Some parameters can be specified by the user. In the generated 3D graphics window, the user can right- or left-click mouse to zoom in or zoom out the 3D graphics, or mouse-drag the graphics to rotate the 3D graphics, or slide scrollbar to vertically or horizontally translate the 3D graphics. The 3D graphics can be saved into an image file (in BMP format). Both molVisual3D and demonstration data files were given.

Keywords 3D visualization; molecules; objects; standalone software; biology; chemistry.

Network Biology
ISSN 2220-8879
URL: http://www.iaees.org/publications/journals/nb/online-version.asp
RSS: http://www.iaees.org/publications/journals/nb/rss.xml
E-mail: networkbiology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

So far, numerous visualization software, e.g., the software for network visualization, have been developed (Narad et al., 2017; Zhang, 2007, 2021, 2023, 2024a-g). Some of them were developed as Java tools, e.g., I developed a Java software for 3D visualization of objects and molecules, based on JDK 1.1 and J2SDK 1.4.2 (Zhang, 2023). Matlab software were also developed for free uses (Zhang, 2021). In the recent, some standalone executable software were developed by using Delphi (Zhang, 2024e-g). Java and Matlab are powerful for developing software tools and thus be widely used. Nevertheless, the JRE and Matlab environment are needed to support these Java and Matlab software respectively. Delphi, which is based on Object Pascal, is the development environment for standalone software. For example, I have developed the standalone executable software for 3D visualization of objects using Delphi (Zhang, 2024e). In present study, I developed a standalone executable software for visualization of molecules. It uses the XYZ file of a molecule to generate the 3D graphics. The full Delphi codes, the free software and demonstration data files were given.

2 Software and User Guide

2.1 Software codes

Based on the previous Java software (Zhang, 2023), The standalone executable software, molVisual3D.exe (Version 1.0), was developed by using Delphi (Fig. 1). The following are the full Delphi codes of the software:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Label6: TLabel;
    Label7: TLabel;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  scale,dila: single;
  ballsize, ballcol, maxscroll: Integer;
  scrollverpos, scrollhorpos: integer;


implementation
```

```
uses Unit2, Unit3;

{$R *.dfm}

procedure TForm1.Button2Click(Sender: TObject);
begin
Application.Terminate;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
ballsize:=strtoint(edit1.Text);
ballcol:=strtoint(edit2.Text);
scale:=strtofloat(edit6.Text);
dila:=strtofloat(edit7.Text);
maxscroll:=strtoint(edit3.Text);
form2.width:=strtoint(edit4.Text);
form2.height:=strtoint(edit5.Text);
form2.scrollbar1.max:=maxscroll;
form2.scrollbar2.max:=maxscroll;
scrollverpos:=0;
scrollhorpos:=0;
form1.visible:=false;
form2.visible:=true;
Application.BringToFront;
end;

end.



unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, math, ExtCtrls, ExtDlgs, StrUtils, Unit3;

type
  TForm2 = class(TForm)
    SavePictureDialog1: TSavePictureDialog;
    Button1: TButton;
    OpenDialog1: TOpenDialog;
    PaintBox1: TPaintBox;
    ScrollBar1: TScrollBar;
    ScrollBar2: TScrollBar;
    procedure Button1Click(Sender: TObject);
    procedure PaintBox1MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormShow(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure PaintBox1DragOver(Sender, Source: TObject; X, Y: Integer;
      State: TDragState; var Accept: Boolean);
    procedure PaintBox1DragDrop(Sender, Source: TObject; X, Y: Integer);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure ScrollBar1Change(Sender: TObject);
    procedure ScrollBar2Change(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
   Form2: TForm2;
   xx, xy, xz, xo, yx, yy, yz, yo, zx, zy, zz, zo: single;
   axx, axy, axz, axo, ayx, ayy, ayz, ayo, azx, azy, azz, azo: single;
   txx, txy, txz, txo, tyx, tyy, tyz, tyo, tzx, tzy, tzz, tzo: single;
   xmin, xmax, ymin, ymax, zmin, zmax: single;
   xfa, xfac, xfac0: single;
   nvert, maxvert: integer;
   vert: array of single;
   tvert: array of integer;
   prevx, prevy: integer;
   transformed: boolean;
   filename: string;
   res: array of string;
   atoms: array of Atom;
   defaultAtom: Atom;
   mres, lab, maxr: integer;
   difx, dify, poshor, posver, scrollverpos, scrollhorpos: integer;
   dataColor: array of shortint;

implementation

uses Unit1;

{$R *.dfm}
```

```
procedure Mat3D();
begin
xx:=1.0;
yy:=1.0;
zz:=1.0;
end;

procedure amatMat3D();
begin
axx:=1.0;
ayy:=1.0;
azz:=1.0;
end;

procedure tmatMat3D();
begin
txx:=1.0;
tyy:=1.0;
tzz:=1.0;
end;

procedure scalef(f: single);
begin
xx:=xx*f;
xy:=xy*f;
xz:=xz*f;
xo:=xo*f;
yx:=yx*f;
yy:=yy*f;
yz:=yz*f;
yo:=yo*f;
zx:=zx*f;
zy:=zy*f;
zz:=zz*f;
zo:=zo*f;
end;

procedure scalefff(f: single; f1: single; f2: single);
begin
xx:=xx*f;
xy:=xy*f;
xz:=xz*f;
xo:=xo*f;
yx:=yx*f1;
```

```
yy:=yy*f1;
yz:=yz*f1;
yo:=yo*f1;
zx:=zx*f2;
zy:=zy*f2;
zz:=zz*f2;
zo:=zo*f2;
end;

procedure translate(f: single; f1: single; f2: single);
begin
xo:=xo+f;
yo:=yo+f1;
zo:=zo+f2;
end;

procedure amattranslate(f: single; f1: single; f2: single);
begin
axo:=axo+f;
ayo:=ayo+f1;
azo:=azo+f2;
end;

procedure xrot(d: single);
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=yx*d1+zx*d2;
f1:=yy*d1+zy*d2;
f2:=yz*d1+zz*d2;
f3:=yo*d1+zo*d2;
f4:=zx*d1-yx*d2;
f5:=zy*d1-yy*d2;
f6:=zz*d1-yz*d2;
f7:=zo*d1-yo*d2;
yo:=f3;
yx:=f;
yy:=f1;
yz:=f2;
zo:=f7;
zx:=f4;
```

```
zy:=f5;
zz:=f6;
end;

procedure yrot(d: single);
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=xx*d1+zx*d2;
f1:=xy*d1+zy*d2;
f2:=xz*d1+zz*d2;
f3:=xo*d1+zo*d2;
f4:=zx*d1-xx*d2;
f5:=zy*d1-xy*d2;
f6:=zz*d1-xz*d2;
f7:=zo*d1-xo*d2;
xo:=f3;
xx:=f;
xy:=f1;
xz:=f2;
zo:=f7;
zx:=f4;
zy:=f5;
zz:=f6;
end;

procedure zrot(d: single);
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=yx*d1+xx*d2;
f1:=yy*d1+xy*d2;
f2:=yz*d1+xz*d2;
f3:=yo*d1+xo*d2;
f4:=xx*d1-yx*d2;
f5:=xy*d1-yy*d2;
f6:=xz*d1-yz*d2;
```

```
f7:=xo*d1-yo*d2;
yo:=f3;
yx:=f;
yy:=f1;
yz:=f2;
xo:=f7;
xx:=f4;
xy:=f5;
xz:=f6;
end;

procedure amatxrot(d: single);
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=ayx*d1+azx*d2;
f1:=ayy*d1+azy*d2;
f2:=ayz*d1+azz*d2;
f3:=ayo*d1+azo*d2;
f4:=azx*d1-ayx*d2;
f5:=azy*d1-ayy*d2;
f6:=azz*d1-ayz*d2;
f7:=azo*d1-ayo*d2;
ayo:=f3;
ayx:=f;
ayy:=f1;
ayz:=f2;
azo:=f7;
azx:=f4;
azy:=f5;
azz:=f6;
end;

procedure amatyrot(d: single);
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
```

```
f:=axx*d1+azx*d2;
f1:=axy*d1+azy*d2;
f2:=axz*d1+azz*d2;
f3:=axo*d1+azo*d2;
f4:=azx*d1-axx*d2;
f5:=azy*d1-axy*d2;
f6:=azz*d1-axz*d2;
f7:=azo*d1-axo*d2;
axo:=f3;
axx:=f;
axy:=f1;
axz:=f2;
azo:=f7;
azx:=f4;
azy:=f5;
azz:=f6;
end;

procedure tmatxrot(d: single);
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=tyx*d1+tzx*d2;
f1:=tyy*d1+tzy*d2;
f2:=tyz*d1+tzz*d2;
f3:=tyo*d1+tzo*d2;
f4:=tzx*d1-tyx*d2;
f5:=tzy*d1-tyy*d2;
f6:=tzz*d1-tyz*d2;
f7:=tzo*d1-tyo*d2;
tyo:=f3;
tyx:=f;
tyy:=f1;
tyz:=f2;
tzo:=f7;
tzx:=f4;
tzy:=f5;
tzz:=f6;
end;

procedure tmatyrot(d: single);
```

```
var
   d1,d2: single;
   f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=txx*d1+tzx*d2;
f1:=txy*d1+tzy*d2;
f2:=txz*d1+tzz*d2;
f3:=txo*d1+tzo*d2;
f4:=tzx*d1-txx*d2;
f5:=tzy*d1-txy*d2;
f6:=tzz*d1-txz*d2;
f7:=tzo*d1-txo*d2;
txo:=f3;
txx:=f;
txy:=f1;
txz:=f2;
tzo:=f7;
tzx:=f4;
tzy:=f5;
tzz:=f6;
end;

procedure mult(xx1: single; xy1: single; xz1: single; xo1: single; yx1: single; yy1: single; yz1: single; yo1: single; zx1: single; zy1: single; zz1: single; zo1: single);
var
   f, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11: single;
begin
f:=xx*xx1+yx*xy1+zx*xz1;
f1:=xy*xx1+yy*xy1+zy*xz1;
f2:=xz*xx1+yz*xy1+zz*xz1;
f3:=xo*xx1+yo*xy1+zo*xz1+xo1;
f4:=xx*yx1+yx*yy1+zx*yz1;
f5:=xy*yx1+yy*yy1+zy*yz1;
f6:=xz*yx1+yz*yy1+zz*yz1;
f7:=xo*yx1+yo*yy1+zo*yz1+yo1;
f8:=xx*zx1+yx*zy1+zx*zz1;
f9:=xy*zx1+yy*zy1+zy*zz1;
f10:=xz*zx1+yz*zy1+zz*zz1;
f11:=xo*zx1+yo*zy1+zo*zz1+zo1;
xx:=f;
xy:=f1;
xz:=f2;
```

```
xo:=f3;
yx:=f4;
yy:=f5;
yz:=f6;
yo:=f7;
zx:=f8;
zy:=f9;
zz:=f10;
zo:=f11;
end;

procedure amatmult(xx1: single; xy1: single; xz1: single; xo1: single; yx1: single; yy1: single; yz1: single; yo1: single; zx1:
single; zy1: single; zz1: single; zo1: single);
var
   f, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11: single;
begin
f:=axx*xx1+ayx*xy1+azx*xz1;
f1:=axy*xx1+ayy*xy1+azy*xz1;
f2:=axz*xx1+ayz*xy1+azz*xz1;
f3:=axo*xx1+ayo*xy1+azo*xz1+xo1;
f4:=axx*yx1+ayx*yy1+azx*yz1;
f5:=axy*yx1+ayy*yy1+azy*yz1;
f6:=axz*yx1+ayz*yy1+azz*yz1;
f7:=axo*yx1+ayo*yy1+azo*yz1+yo1;
f8:=axx*zx1+ayx*zy1+azx*zz1;
f9:=axy*zx1+ayy*zy1+azy*zz1;
f10:=axz*zx1+ayz*zy1+azz*zz1;
f11:=axo*zx1+ayo*zy1+azo*zz1+zo1;
axx:=f;
axy:=f1;
axz:=f2;
axo:=f3;
ayx:=f4;
ayy:=f5;
ayz:=f6;
ayo:=f7;
azx:=f8;
azy:=f9;
azz:=f10;
azo:=f11;
end;

procedure transmat(var af: array of single; var ai: array of integer; nvert: integer);
var
   j: integer;
```

```
    f, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14: single;
begin
f:=xx;
f1:=xy;
f2:=xz;
f3:=xo;
f4:=yx;
f5:=yy;
f6:=yz;
f7:=yo;
f8:=zx;
f9:=zy;
f10:=zz;
f11:=zo;
j:=nvert*3;
while(j>=0) do
begin
f12:=af[j];
f13:=af[j+1];
f14:=af[j+2];
ai[j]:=floor(f12*f+f13*f1+f14*f2+f3);
ai[j+1]:=floor(f12*f4+f13*f5+f14*f6+f7);
ai[j+2]:=floor(f12*f8+f13*f9+f14*f10+f11);
j:=j-3;
end;
end;

procedure units();
begin
xo:=0.0;
xx:=1.0;
xy:=0.0;
xz:=0.0;
yo:=0.0;
yx:=0.0;
yy:=1.0;
yz:=0.0;
zo:=0.0;
zx:=0.0;
zy:=0.0;
zz:=1.0;
end;

procedure tmatunits();
begin
```

```
txo:=0.0;
txx:=1.0;
txy:=0.0;
txz:=0.0;
tyo:=0.0;
tyx:=0.0;
tyy:=1.0;
tyz:=0.0;
tzo:=0.0;
tzx:=0.0;
tzy:=0.0;
tzz:=1.0;
end;

procedure staticColor();
var
   i, j, k, l, i1, j1, k1, l1: integer;
begin
setLength(dataColor,6400);
i:=0;
for j:=79 downto 0 do
begin
k:=floor(sqrt(1600-(j-40)*(j-40))+0.5);
l:=(j*80+40)-k;
i1:=-k;
while(i1<k) do
begin
j1:=i1+15;
k1:=(j-40)+15;
l1:=floor(sqrt(j1*j1+k1*k1)+0.5);
if(l1>i) then
i:=l1;
if(l1>0) then
begin
l:=l+1;
dataColor[l]:=shortint(l1);
end
else
begin
l:=l+1;
dataColor[l]:=1;
end;
i1:=i1+1;
end;
end;
```

```
maxr:=i;
end;

function atomTable(s: string): Atom;
var
   atm: Atom;
begin
if((pos('c',s)<>0) or (pos('C',s)<>0)) then
begin
atm.Atom(210,190,80); result:=atm; exit;
end
else if((pos('h',s)<>0) or (pos('H',s)<>0)) then
begin
atm.Atom(120,250,90); result:=atm; exit;
end
else if((pos('n',s)<>0) or (pos('N',s)<>0)) then
begin
atm.Atom(170,100,20); result:=atm; exit;
end
else if((pos('o',s)<>0) or (pos('O',s)<>0)) then
begin
atm.Atom(190,100,20); result:=atm; exit;
end
else if((pos('p',s)<>0) or (pos('P',s)<>0)) then
begin
atm.Atom(150,120,20); result:=atm; exit;
end
else if((pos('s',s)<>0) or (pos('S',s)<>0)) then
begin
atm.Atom(140,90,50); result:=atm; exit;
end
else if((pos('si',s)<>0) or (pos('Si',s)<>0) or (pos('SI',s)<>0)) then
begin
atm.Atom(100,180,150); result:=atm; exit;
end
else if((pos('b',s)<>0) or (pos('B',s)<>0)) then
begin
atm.Atom(210,100,70); result:=atm; exit;
end
else if((pos('i',s)<>0) or (pos('I',s)<>0)) then
begin
atm.Atom(110,90,30); result:=atm; exit;
end
else if((pos('cl',s)<>0) or (pos('Cl',s)<>0) or (pos('CL',s)<>0)) then
begin
```

```
atm.Atom(180,120,10); result:=atm; exit;
end
else if((pos('br',s)<>0) or (pos('Br',s)<>0) or (pos('BR',s)<>0)) then
begin
atm.Atom(80,180,190); result:=atm; exit;
end
else if((pos('fe',s)<>0) or (pos('Fe',s)<>0) or (pos('FE',s)<>0)) then
begin
atm.Atom(15,230,250); result:=atm; exit;
end
else if((pos('ca',s)<>0) or (pos('Ca',s)<>0) or (pos('CA',s)<>0)) then
begin
atm.Atom(10,195,235); result:=atm; exit;
end
else if((pos('mg',s)<>0) or (pos('Mg',s)<>0) or (pos('MG',s)<>0)) then
begin
atm.Atom(85,200,240); result:=atm; exit;
end
else if((pos('k',s)<>0) or (pos('K',s)<>0)) then
begin
atm.Atom(80,190,210); result:=atm; exit;
end
else if((pos('na',s)<>0) or (pos('Na',s)<>0) or (pos('NA',s)<>0)) then
begin
atm.Atom(10,140,230); result:=atm; exit;
end
else if((pos('sr',s)<>0) or (pos('Sr',s)<>0) or (pos('SR',s)<>0)) then
begin
atm.Atom(10,200,220); result:=atm; exit;
end
else if((pos('zn',s)<>0) or (pos('Zn',s)<>0) or (pos('ZN',s)<>0)) then
begin
atm.Atom(10,90,190); result:=atm; exit;
end
else if((pos('cu',s)<>0) or (pos('Cu',s)<>0) or (pos('CU',s)<>0)) then
begin
atm.Atom(50,110,200); result:=atm; exit;
end
else if((pos('cr',s)<>0) or (pos('Cr',s)<>0) or (pos('CR',s)<>0)) then
begin
atm.Atom(20,110,210); result:=atm; exit;
end
else if((pos('se',s)<>0) or (pos('Se',s)<>0) or (pos('SE',s)<>0)) then
begin
atm.Atom(50,200,230); result:=atm; exit;
```

```
end
else if((pos('mo',s)<>0) or (pos('Mo',s)<>0) or (pos('MO',s)<>0)) then
begin
atm.Atom(30,100,140); result:=atm; exit;
end
else if((pos('mn',s)<>0) or (pos('Mn',s)<>0) or (pos('MN',s)<>0)) then
begin
atm.Atom(40,90,160); result:=atm; exit;
end
else if((pos('co',s)<>0) or (pos('Co',s)<>0) or (pos('CO',s)<>0)) then
begin
atm.Atom(30,95,180); result:=atm; exit;
end
else if((pos('hn',s)<>0) or (pos('HN',s)<>0)) then
begin
atm.Atom(130,130,130); result:=atm; exit;
end
else lab:=1;
end;

procedure Model3D();
begin
Mat3D();
xrot(20.0);
yrot(30.0);
end;

procedure addVert(s: string; f: single; f1: single; f2: single);
var
   i: integer;
   axtom: Atom;
begin
i:=nvert;
if(length(vert)=0) then
begin
maxvert:=100000;
setLength(vert,maxvert*6);
setLength(atoms,maxvert);
end
else
begin
axtom:=atomTable(s);
if(lab=1) then
begin
defaultAtom.Atom(255,100,200);
```

```
axtom:=defaultAtom;
lab:=0;
end;
atoms[i]:=axtom;
end;
i:=i*3;
vert[i]:=f;
vert[i+1]:=f1;
vert[i+2]:=f2;
nvert:=nvert+1;
end;

procedure transform();
begin
if((transformed=true) or (nvert=0)) then
exit;
if(length(tvert)<nvert*6) then
setLength(tvert,nvert*6);
transmat(vert,tvert,nvert);
transformed:=true;
end;

procedure findBB();
var
   f, f1, f2, f3, f4, f5, f6, f7, f8: single;
   i, j: integer;
   af: array of single;
begin
if(nvert<=0) then
exit;
setLength(af,length(vert));
for j:=0 to length(vert)-1 do
af[j]:=vert[j];
f:=af[0];
f1:=f;
f2:=af[1];
f3:=f2;
f4:=af[2];
f5:=f4;
i:=nvert*3;
while(true) do
begin
i:=i-3;
if(i<=0) then
break;
```

```
f6:=af[i];
if(f6<f) then
f:=f6;
if(f6>f1) then
f1:=f6;
f7:=af[i+1];
if(f7<f2) then
f2:=f7;
if(f7>f3) then
f3:=f7;
f8:=af[i+2];
if(f8<f4) then
f4:=f8;
if(f8>f5) then
f5:=f8;
end;
af:=nil;
xmax:=f1;
xmin:=f;
ymax:=f3;
ymin:=f2;
zmax:=f5;
zmin:=f4;
end;

procedure modelpaint();
var
   i, i1, j1, j2, k1, l1, l2, i3, i4, k3: integer;
   ai, ai1: array of integer;
   flag: boolean;
begin
if((length(vert)=0) or (nvert=0)) then
exit;
transform();
setLength(ai,length(tvert));
for i:=0 to length(tvert)-1 do
ai[i]:=tvert[i];
if(length(ai1)=0) then
begin
setLength(ai1,nvert);
for i1:=nvert-1 downto 0 do
ai1[i1]:=i1*3;
end;
j1:=nvert-1;
flag:=false;
```

```
repeat
j1:=j1-1;
if(j1<0) then
break;
flag:=false;
for k1:=0 to j1 do
begin
j2:=ai1[k1];
l2:=ai1[k1+1];
if(ai[j2+2]>ai[l2+2]) then
begin
ai1[k1+1]:=j2;
ai1[k1]:=l2;
flag:=true;
end;
end;
until(flag=false);
l1:=nvert;
if((l1<=0) or (nvert=0)) then
exit;
for i3:=0 to l1-1 do
begin
k3:=ai1[i3];
i4:=ai[k3+2];
if(i4<0) then
i4:=0;
if(i4>15) then
i4:=15;
atoms[round(k3/3.0)].paint(ai[k3],ai[k3+1],i4,atoms[round(k3/3.0)].Rl,atoms[round(k3/3.0)].Gl,atoms[round(k3/3.0)].Bl,bal
lcol,ballsize);
end;
for i:=0 to length(ai)-1 do
tvert[i]:=ai[i];
end;

procedure appletpaint();
begin
units();
translate(-(xmin+xmax)/2.0,-(ymin+ymax)/2.0,-(zmin+zmax)/2.0);
mult(axx,axy,axz,axo,ayx,ayy,ayz,ayo,azx,azy,azz,azo);
scalefff(xfac,-xfac,(16.0*xfac)/form2.width);
translate(form2.width/2.0,form2.height/2.0,8.0);
transformed:=false;
modelpaint();
end;
```

```
procedure strval(s: string);
var
    str: Tstringlist;
    i: integer;
begin
s:=trim(s);
str:=TStringList.Create;
try
str.Delimiter:=' ';
str.DelimitedText:=s;
mres:=str.count;
setLength(res,mres);
for i:=0 to mres-1 do
res[i]:=str[i];
finally
str.Free;
end;
end;

procedure run();
var
    f, f1, f2, f3, f4, sig: single;
    Fil: textfile;
    s: string;
begin
amatMat3D();
tmatMat3D();
amatxrot(20.0);
amatyrot(20.0);
Model3D();
maxvert:=0;
nvert:=0;
AssignFile(Fil,filename);
Reset(Fil);
while not Eof(Fil) do
begin
Readln(Fil,s);
s:=trim(s);
if((pos('#',s)=1) or (s='')) then
continue;
strval(s);
if(mres=4) then
addVert(res[0],strtofloat(res[1]),strtofloat(res[2]),strtofloat(res[3]));
end;
```

```
CloseFile(Fil);
res:=nil;
findBB();
f:=xmax-xmin;
f1:=ymax-ymin;
f2:=zmax-zmin;
if(f1>f) then
f:=f1;
if(f2>f) then
f:=f2;
f3:=form2.width/f;
f4:=form2.height/f;
if(f3<f4) then sig:=f3
else sig:=f4;
xfac:=0.7*sig*scale;
xfac0:=xfac;
xfa:=xfac/xfac0;
staticColor();
appletpaint();
end;

procedure TForm2.Button1Click(Sender: TObject);
var
   str: string;
   bmp: TBitMap;
   R: TRect;
begin
bmp:=TBitmap.Create;
bmp.Width:=PaintBox1.Width;
bmp.Height:=PaintBox1.Height;
R:=Rect(Paintbox1.Left,Paintbox1.Top,Paintbox1.Width,Paintbox1.Height);
bmp.Canvas.CopyRect(R,PaintBox1.Canvas,R);
if savePictureDialog1.execute then
filename:=savePictureDialog1.filename;
scrollbar1.visible:=false;
scrollbar2.visible:=false;
str:=ExtractFileExt(filename);
bmp.SaveToFile(filename);
bmp.Free;
scrollbar1.visible:=true;
scrollbar2.visible:=true;
form2.PaintBox1.refresh;
end;

procedure TForm2.PaintBox1MouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
xfa:=xfac/xfac0;
prevx:=floor(X*xfa);
prevy:=floor(Y*xfa);
if((Button=mbright) or (Button=mbmiddle)) then
if((1.0-dila)>0.0) then
xfac:=xfac-xfac*dila
else
xfac:=0.1;
form2.PaintBox1.Repaint;
appletpaint();
end;

procedure TForm2.PaintBox1DragDrop(Sender, Source: TObject; X, Y: Integer);
begin
xfa:=xfac/xfac0;
prevx:=floor(X*xfa);
prevy:=floor(Y*xfa);
xfac:=xfac+xfac*dila;
form2.PaintBox1.Repaint;
appletpaint();
end;

procedure TForm2.PaintBox1DragOver(Sender, Source: TObject; X, Y: Integer;
    State: TDragState; var Accept: Boolean);
var
     i, j: integer;
     f, f1: single;
begin
i:=floor(X*xfa);
j:=floor(Y*xfa);
tmatunits();
f:=(prevy-j)*(360.0/(form2.width*xfa));
f1:=(i-prevx)*(360.0/(form2.height*xfa));
tmatxrot(f);
tmatyrot(f1);
amatmult(txx,txy,txz,txo,tyx,tyy,tyz,tyo,tzx,tzy,tzz,tzo);
form2.PaintBox1.Repaint;
appletpaint();
prevx:=i;
prevy:=j;
end;

procedure TForm2.FormShow(Sender: TObject);
```

```
begin
if opendialog1.execute then
filename:=OpenDialog1.filename;
run();
end;

procedure TForm2.FormResize(Sender: TObject);
begin
form2.PaintBox1.Repaint;
appletpaint();
end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Application.Terminate;
end;

procedure TForm2.ScrollBar1Change(Sender: TObject);
begin
poshor:=ScrollBar1.position;
difx:=poshor-scrollhorpos;
scrollhorpos:=poshor;
amattranslate(-difx,0.0,0.0);
form2.PaintBox1.Repaint;
appletpaint();
end;

procedure TForm2.ScrollBar2Change(Sender: TObject);
begin
posver:=ScrollBar2.position;
dify:=posver-scrollverpos;
scrollverpos:=posver;
amattranslate(0.0,dify,0.0);
form2.PaintBox1.Repaint;
appletpaint();
end;

end.



unit Unit3;

interface

uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, math, ExtCtrls, ExtDlgs;

type
  Atom = Object
    procedure Atom(i: integer; j: integer; k: integer);
    function blendColor(i: integer; j: integer; f: single): integer;
    procedure createColor();
    procedure paint(i: integer; j: integer; k: integer; rll: integer; gll: integer; bll: integer; ballco: integer; ballsi: integer);

  private
    { Private declarations }
  public
    Rl, Gl, Bl: integer;

    { Public declarations }
  end;

var
     ballcol, ballsize, RRl, GGl, BBl: integer;
     abyte0, abyte1, abyte2: array of shortint;

implementation

uses Unit2;

procedure Atom.Atom(i: integer; j: integer; k: integer);
begin
Rl:=i;
Gl:=j;
Bl:=k;
end;

function Atom.blendColor(i: integer; j: integer; f: single): integer;
begin
result:=floor(j+(i-j)*f);
end;

procedure Atom.createColor();
var
   i, j: integer;
   f, f1: single;
begin
setLength(abyte0,256);
setLength(abyte1,256);
setLength(abyte2,256);
for i:=0 to ballcol-1 do
begin
f:=(i+1.0)/ballcol;
for j:=maxr downto 1 do
begin
f1:=j/maxr;
abyte0[j]:=shortint(blendColor(blendColor(RRl,255,f1),150,f));
abyte1[j]:=shortint(blendColor(blendColor(GGl,255,f1),150,f));
abyte2[j]:=shortint(blendColor(blendColor(BBl,255,f1),150,f));
end;
end;
end;

procedure Atom.paint(i: integer; j: integer; k: integer; rll: integer; gll: integer; bll: integer; ballco: integer; ballsi: integer);
var
   R: TRect;
   l: integer;
begin
ballcol:=ballco;
ballsize:=ballsi;
RRl:=rll;
GGl:=gll;
BBl:=bll;
createColor();
l:=floor(round(ballsize+0.5))+k;
R:=rect(i-(l shr 1),j-(l shr 1),i-(l shr 1)+round(ballsize),j-(l shr 1)+round(ballsize));
if(TColor(RGB(abyte0[k],abyte1[k],abyte2[k]))<>clblack) then
form2.PaintBox1.Canvas.Brush.color:=TColor(RGB(abyte0[k],abyte1[k],abyte2[k]))
else
form2.PaintBox1.Canvas.Brush.color:=clwhite;
form2.PaintBox1.Canvas.Ellipse(R);
end;

end.
```

2.2 User guide

The data files for 3D visualization of molecules are XYZ files (*.xyz). XYZ is a file format (*.xyz) for recording structures of chemical molecules and other objects. It is a type of text files, which can be opened with a text editor (e.g., notepad) (Fig. 2). Almost all 3D software for molecular visualization supports XYZ files.

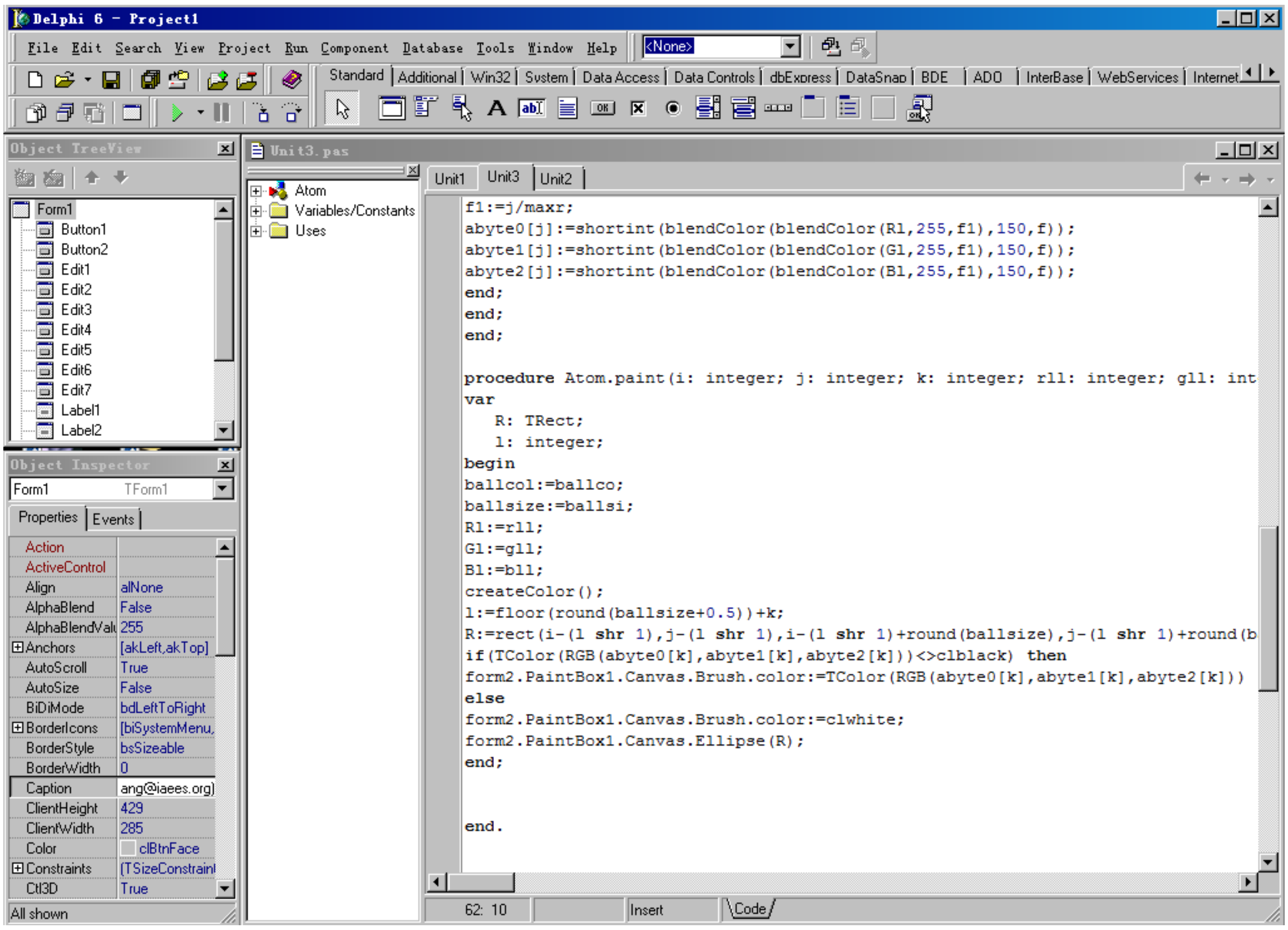
Double-click molVisual3D.exe to run the software, input some parameters (Fig. 3), and choose a data file, the window for 3D graphics will be generated. In the generated 3D graphics window, users may right- or left-click mouse to zoom in or zoom out the 3D graphics, or mouse-drag the graphics to rotate the 3D graphics, or slide scrollbar to vertically or horizontally translate the 3D graphics. Finally, the 3D graphics can be saved as an image file (in BMP format) at its current appearance.

Users may download free XYZ files from internet resources for using the present software. There are numerous XYZ files on the internet.

In addition to molecules, any objects represented by XYZ files can be visualized with the software also.

The software and demo data files (demo data were introduced from JDK 1.1, etc) are included in the package:
http://www.iaees.org/publications/journals/nb/articles/2024-14(2)/e-suppl/Zhang-Supplementary-Material.rar

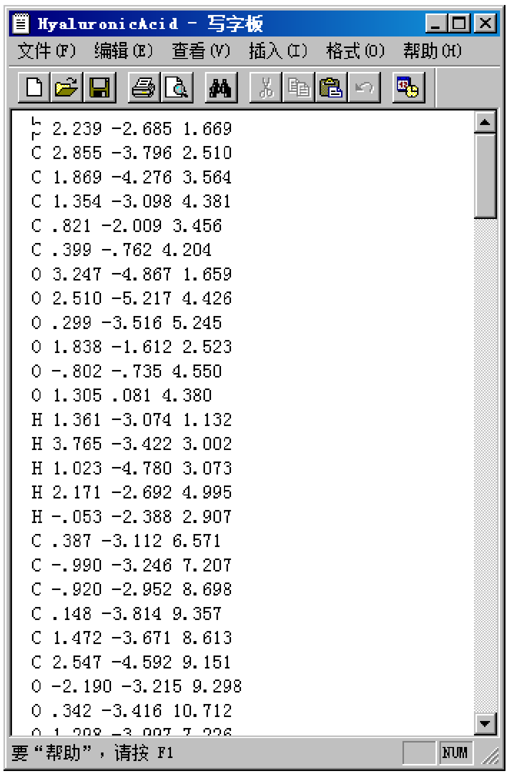
Users may occasionally examine and download available higher versions of the software in this package.



Action	
ActiveControl	
Align	alNone
AlphaBlend	False
AlphaBlendVal	255
⊞Anchors	[akLeft,akTop]
AutoScroll	True
AutoSize	False
BiDiMode	bdLeftToRight
⊞BorderIcons	[biSystemMenu,
BorderStyle	bsSizeable
BorderWidth	0
Caption	ang@iaees.org)
ClientHeight	429
ClientWidth	285
Color	clBtnFace
⊞Constraints	(TSizeConstraint
Ctl3D	True

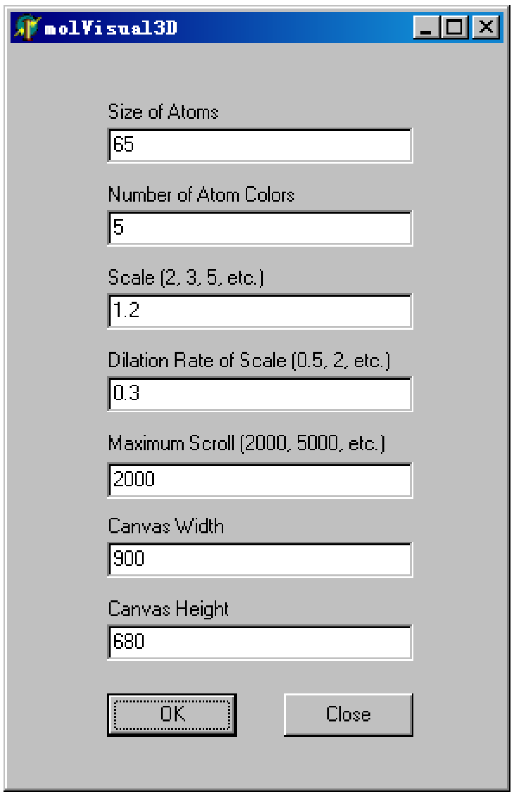
```
f1:=j/maxr;
abyte0[j]:=shortint(blendColor(blendColor(Rl,255,f1),150,f));
abyte1[j]:=shortint(blendColor(blendColor(Gl,255,f1),150,f));
abyte2[j]:=shortint(blendColor(blendColor(Bl,255,f1),150,f));
end;
end;
end;

procedure Atom.paint(i: integer; j: integer; k: integer; rll: integer; gll: int
var
   R: TRect;
   l: integer;
begin
ballcol:=ballco;
ballsize:=ballsi;
Rl:=rll;
Gl:=gll;
Bl:=bll;
createColor();
l:=floor(round(ballsize+0.5))+k;
R:=rect(i-(l shr 1),j-(l shr 1),i-(l shr 1)+round(ballsize),j-(l shr 1)+round(b
if(TColor(RGB(abyte0[k],abyte1[k],abyte2[k]))<>clblack) then
form2.PaintBox1.Canvas.Brush.color:=TColor(RGB(abyte0[k],abyte1[k],abyte2[k]))
else
form2.PaintBox1.Canvas.Brush.color:=clwhite;
form2.PaintBox1.Canvas.Ellipse(R);
end;


end.
```

Fig. 1 The Delphi development environment of molVisual3D.



```
[illegible] 2.239 -2.685 1.669
C 2.855 -3.796 2.510
C 1.869 -4.276 3.564
C 1.354 -3.098 4.381
C .821 -2.009 3.456
C .399 -.762 4.204
O 3.247 -4.867 1.659
O 2.510 -5.217 4.426
O .299 -3.516 5.245
O 1.838 -1.612 2.523
O -.802 -.735 4.550
O 1.305 .081 4.380
H 1.361 -3.074 1.132
H 3.765 -3.422 3.002
H 1.023 -4.780 3.073
H 2.171 -2.692 4.995
H -.053 -2.388 2.907
C .387 -3.112 6.571
C -.990 -3.246 7.207
C -.920 -2.952 8.698
C .148 -3.814 9.357
C 1.472 -3.671 8.613
C 2.547 -4.592 9.151
O -2.190 -3.215 9.298
O .342 -3.416 10.712
O 1.208 -3.097 7.226
```

Fig. 2 A XYZ data file unsed in molVisual3D.



Size of Atoms: 65

Number of Atom Colors: 5

Scale (2, 3, 5, etc.): 1.2

Dilation Rate of Scale (0.5, 2, etc.): 0.3

Maximum Scroll (2000, 5000, etc.): 2000

Canvas Width: 900

Canvas Height: 680

Fig. 3 The parameter input interface of molVisual3D.

3 Demo Examples

The XYZ data for demonstration were from JDK 1.1 (Zhang, 2023). Some of the generated 3D graphics are indicated in Figs 4 and 9.

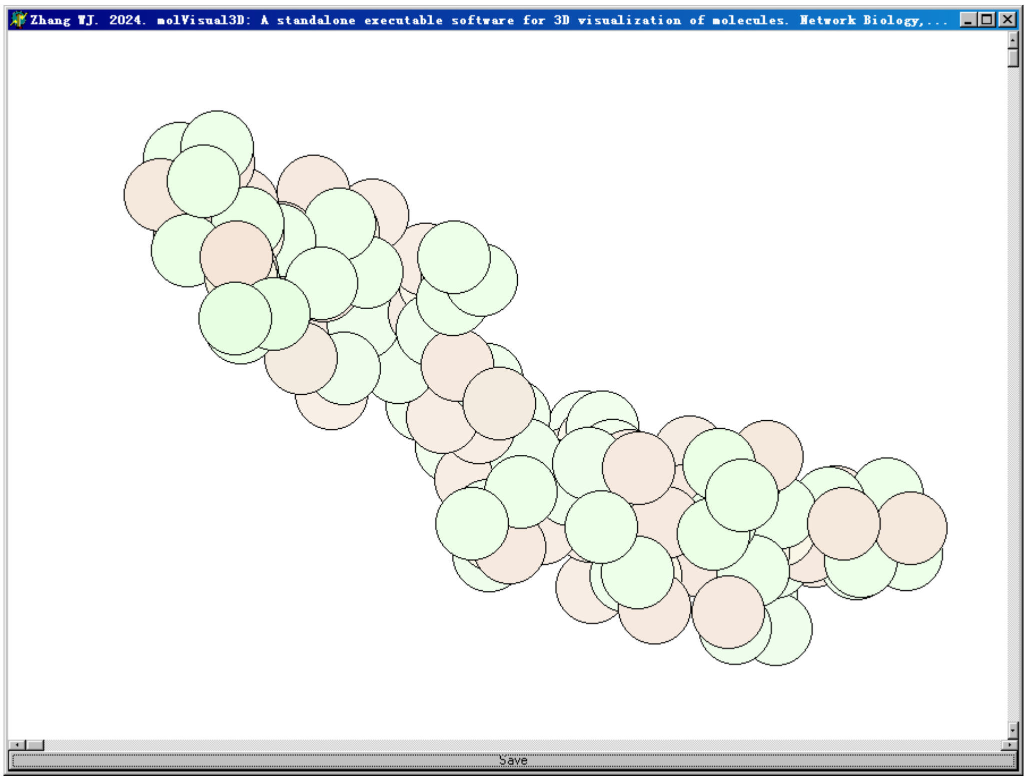

Zhang WJ. 2024. molVisual3D: A standalone executable software for 3D visualization of molecules. Network Biology,...
Save

Fig. 4 The 3D graphics of hyaluronic acid.

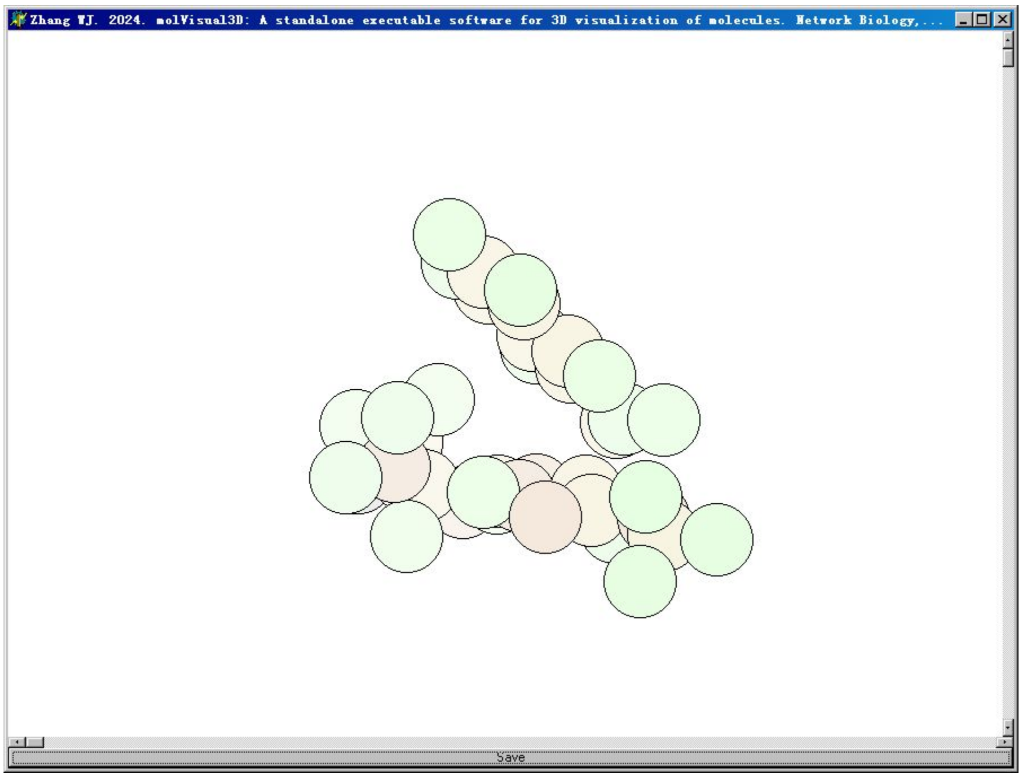

Zhang WJ. 2024. molVisual3D: A standalone executable software for 3D visualization of molecules. Network Biology,...
Save

Fig. 5 The 3D graphics of Aspartame.

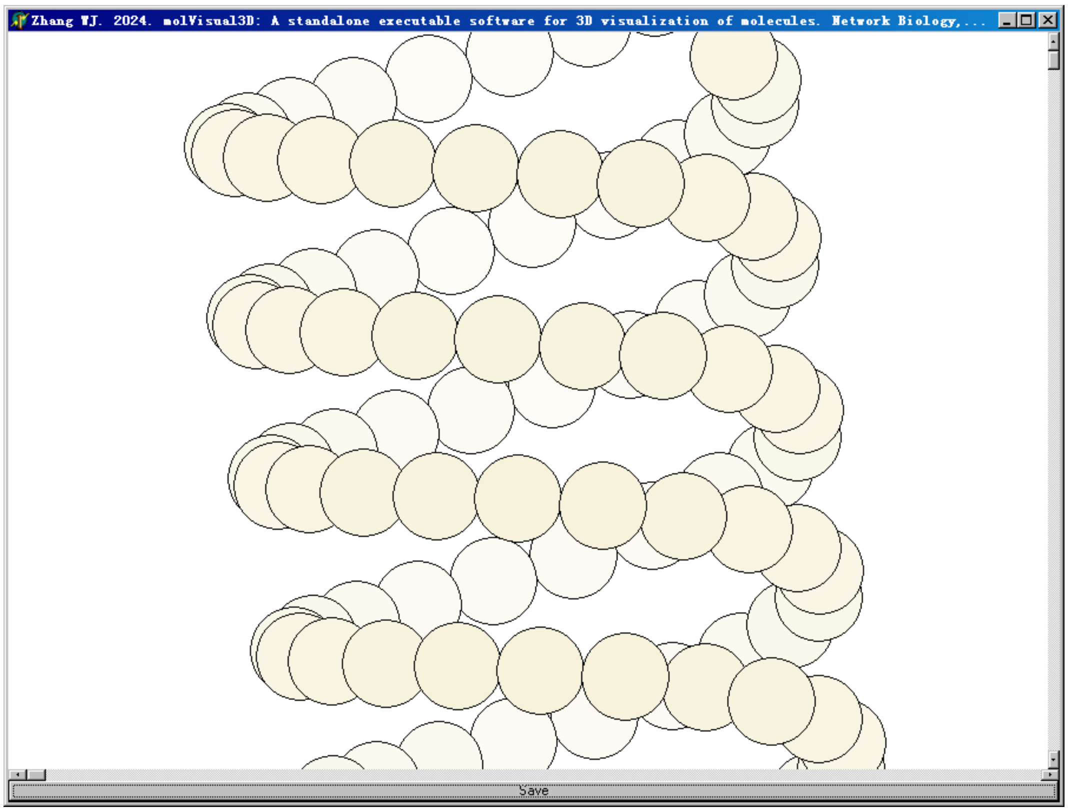

Zhang WJ. 2024. molVisual3D: A standalone executable software for 3D visualization of molecules. Network Biology,...
Save

Fig. 6 The 3D graphics of DNA double helix.

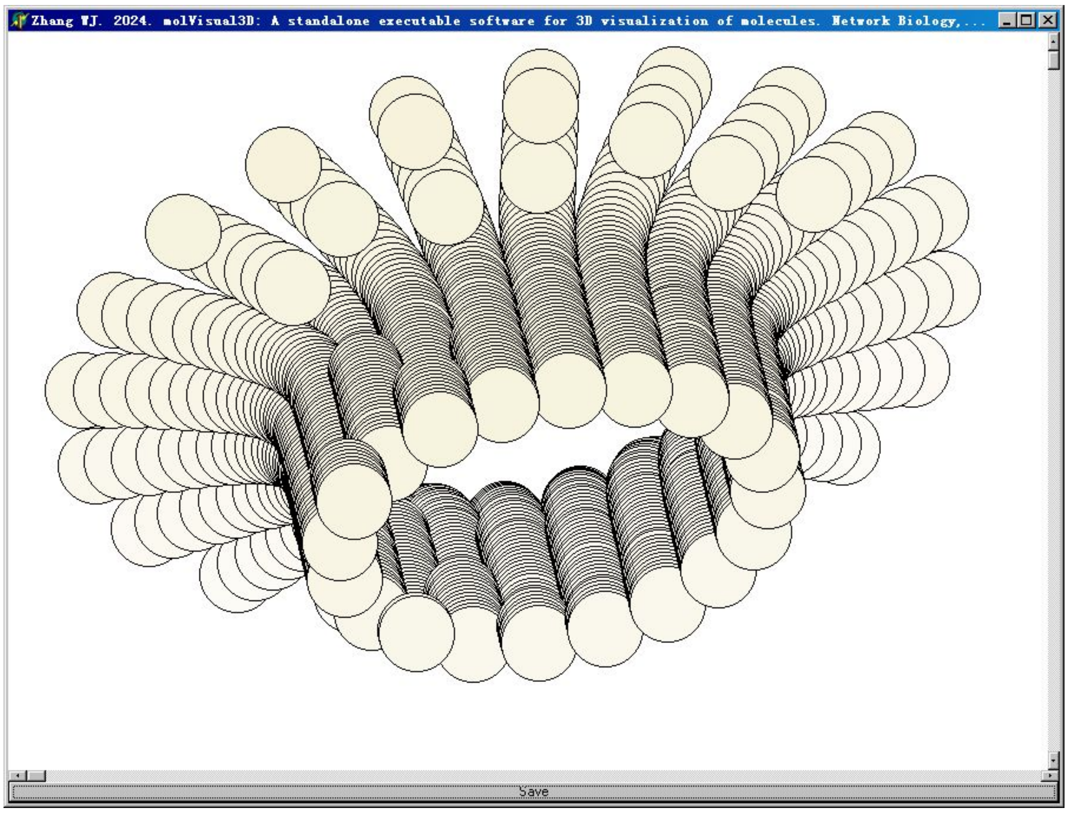

Zhang WJ. 2024. molVisual3D: A standalone executable software for 3D visualization of molecules. Network Biology,...
Save

Fig. 7 The 3D graphics of a mechanical part.

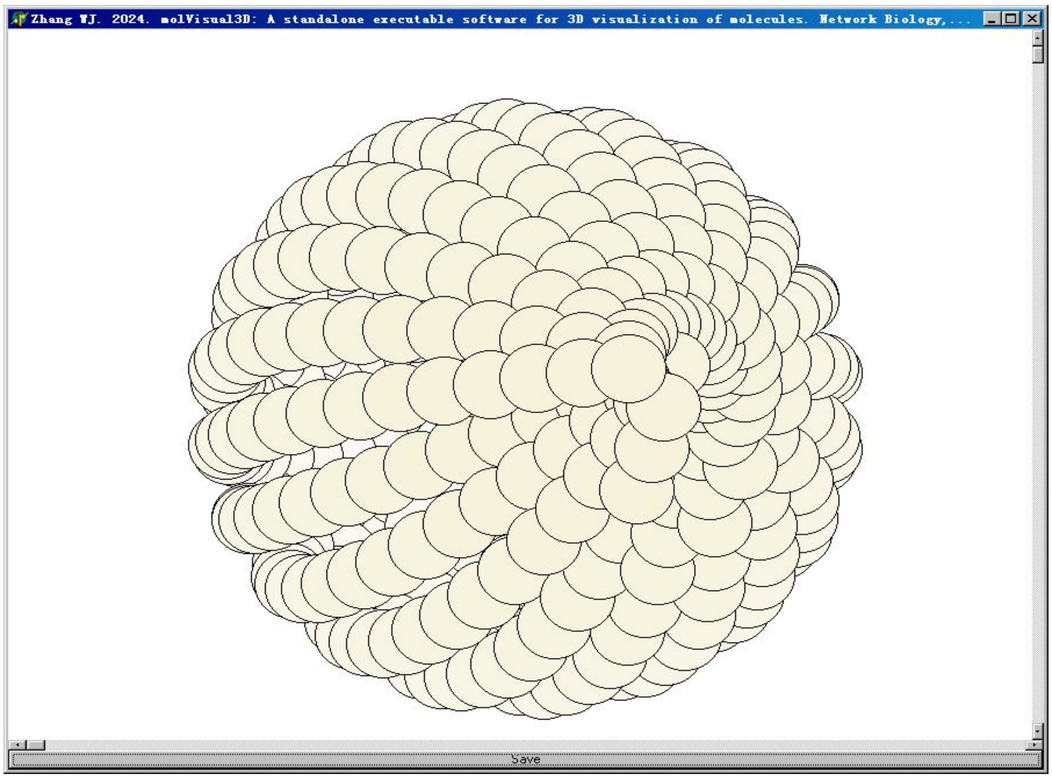

Zhang WJ. 2024. molVisual3D: A standalone executable software for 3D visualization of molecules. Network Biology,...
Save

Fig. 8 The 3D graphics of a sphere.

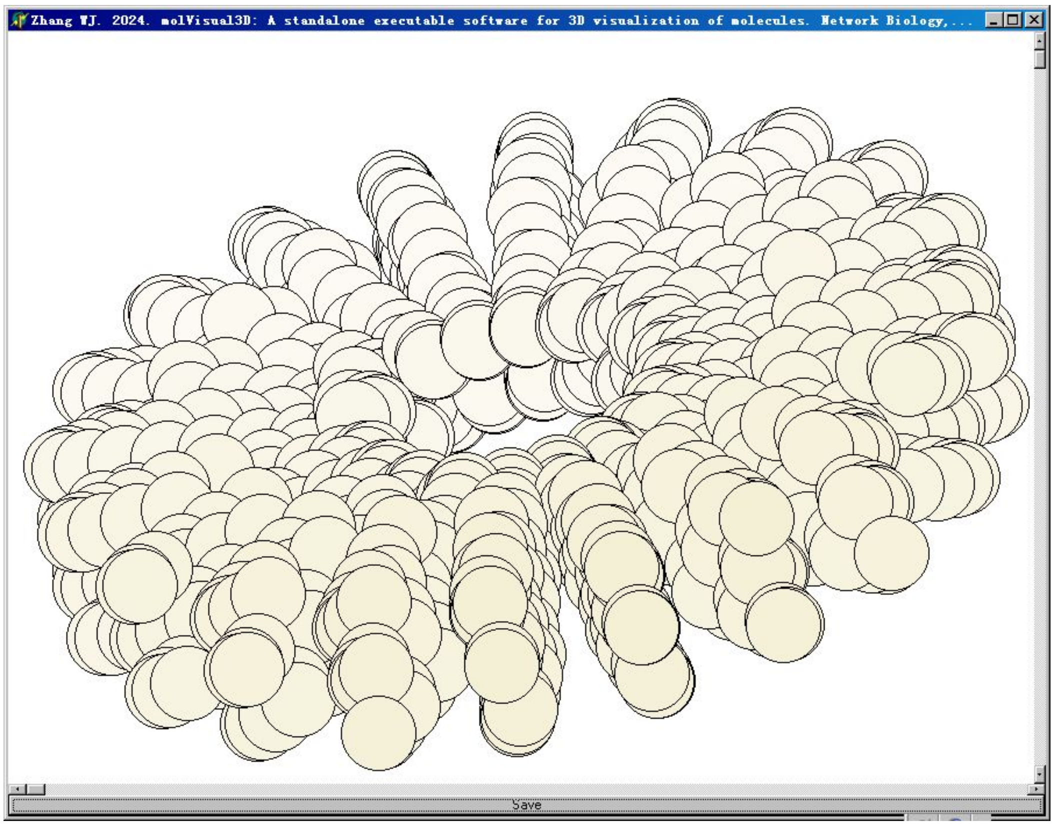

Zhang WJ. 2024. molVisual3D: A standalone executable software for 3D visualization of molecules. Network Biology,...
Save

Fig. 9 The 3D graphics of a mathematical structure.

References

Narad P, Upadhyaya KC, Som A. 2017. Reconstruction, visualization and explorative analysis of human pluripotency network. Network Biology, 7(3): 57-75

Zhang WJ. 2007. Computer inference of network of ecological interactions from sampling data. Environmental Monitoring and Assessment, 124: 253–261

Zhang WJ. 2021. A web tool for generating user-interface interactive networks. Network Biology, 11(4): 247-262

Zhang WJ. 2023. 3D visualization of objects and molecules: An integrative Java software. Computational Ecology and Software, 13(4): 81-108

Zhang WJ. 2024a. A Matlab software for visualizing user-interface interactive networks. Network Biology, 14(1): 13-19

Zhang WJ. 2024b. A standalone executable software for network visualization. Network Pharmacology, 9(1-2): 1-10

Zhang WJ. 2024c. An executable Java software for visualizing networks. Network Biology, 14(1): 1-12

Zhang WJ. 2024d. netGen 3.0: The executable Java software for network visualization. Selforganizology, 11(1-2): 1-27

Zhang WJ. 2024e. objectVisual3D: A standalone executable software for 3D visualization of objects. Selforganizology, 11(3-4): 28-53

Zhang WJ. 2024f. Several digital timers and clocks for desktop computers. Computational Ecology and Software, 14(1): 1-13

Zhang WJ. 2024g. Two image viewers: A projector and a screen saver. Network Pharmacology, 9(3-4): 11-23