Electronic Supplementary Information for

Matrix-localization for fast analysis of arrayed microfluidic immunoassays

Yi Zhang,[‡a,e] Xuwei Wang,[‡b] Lusheng Song,[e] Chuanlai Xu,[c] Liying Ma,[d] Zhanhua Li,[b] Jianzhong Xi*[a] and Xingyu Jiang*[e]

[a] *Department of Biomedical Engineering, College of Engineering, Peking University, Beijing 100871, China. Tel: +86-10-62760698; E-mail: xi@coe.pku.edu.cn*

[b] *State Key Laboratory for Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100080, China*

[c] *School of Food Science and Technology, Jiangnan University, Wuxi 214122, China*

[d] *State Key Laboratory for Infection Disease Prevention and Control, National Center for AIDS/STD Control and Prevention (NCAIDS), Chinese Center for Disease Control and Prevention (China-CDC), Beijing 102206, China*

[e] *CAS Key Lab for Biological Effects of Nanomaterials and Nanosafety, National Center for Nanoscience and Technology, Beijing 100190, China. Tel: +86-10-82545558; Fax: +86-10-82545631; E-mail: xingyujiang@nanoctr.cn*

† Electronic Supplementary Information (ESI) available. See DOI: 10.1039/b000000x/

‡ These authors contributed equally to this work.

Microfluidic immunoassays

(1) Methyl paraoxon immunoassay (Fig. 2A):

Methyl paraoxon conjugated with ovalbumin (MP-OVA) and rabbit antibody specific to MP (against MP-BSA) (R-anti-MP) as 1st antibody were prepared by School of Food Science and Technology at Jiangnan University (Wuxi, Jiangsu, China). We introduced MP-OVA dilutions and negative controls into each channel, and incubated them for 15 min. We peeled off the channels after extraction of solution by a syringe, and used PBST (Phosphate buffered saline with 0.05% (v/v) Tween-20) to wash the surface of substrate 3 times. We blocked the surface with 5% skim milk (Wondersun Inc.) for 8 min, and washed with PBST 3 times. We placed new microfluidic channels on the substrate in the direction perpendicular to the strips of immobilized antigens, and introduced R-anti-MP dilutions and negative controls into each channel, followed by incubation for 15 min. We peeled off the channels after extraction of solution by a syringe, and also used PBST to wash the surface of substrate 3 times. We dripped 1:150 diluted (in PBS) 2nd antibody, goat anti-rabbit IgG (H+L) conjugated with Cy3 (G-anti-R-Cy3) (Beyotime Institute of Biotechnology), on the entire region of immunoreaction, and incubated them for 15 min. Finally, we used PBST to wash the surface of substrate 3 times. We captured the fluorescence image by the microscope (Leica DMI 6000B coupled with a Leica DFC350 FX digital monochrome camera; the excitation filter is band-pass 515-560 nm; the dichromatic mirror is at 580 nm; the emission filter is long-pass 590 nm).

(2) HIV immunoassay (Fig. 3A):

The candidate sera and the HIV surface antigens, p24, p31, gp41, p51, p55, gp120, and gp36, were obtained from Chinese Center for Disease Control and Prevention (China CDC). We introduced the HIV antigens and negative control into each channel, and incubated them for 10 min. We peeled off channels after extraction of solution, and used PBST to wash the surface of substrate 3 times. We placed new channels on the substrate in the direction perpendicular to the strips of immobilized antigens. We introduced different human sera, HIV positive/negative control sera, and 5% albumin bovine serum (BSA) (Merck) as blank control into each channel, and incubated them

for 10 min. We peeled off channels after extraction of solution, and also used PBST to wash the surface of substrate 5 times. We introduced 1:50 diluted (in PBST) 2nd antibody, rabbit anti-human IgG conjugated with FITC (R-anti-H-FITC) (ZhongShan GoldenBridge Biotechnology Co. Ltd.), on the entire region of immunoreaction, and incubated it for 10 min. Finally, we used PBST to wash the surface of substrate 3 times. We captured the fluorescence image by the microscope (the excitation filter is band-pass 450-490 nm; the dichromatic mirror is at 510 nm; the emission filter is long-pass 515 nm).

Matlab programming code

Our program including the GUI environment consists of 12 separate files (10 m-files and 2 fig-files). The code in each m-file is listed bellow:

(1) zy.m

```
function varargout = zy(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',            mfilename, ...
                   'gui_Singleton',    gui_Singleton, ...
                   'gui_OpeningFcn', @zy_OpeningFcn, ...
                   'gui_OutputFcn',    @zy_OutputFcn, ...
                   'gui_LayoutFcn',    [] , ...
                   'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function zy_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = zy_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function file_Callback(hObject, eventdata, handles)
function imopen_Callback(hObject, eventdata, handles)
 [imfile, impath,tag] = uigetfile({'*.tif','TIFF file';'*.*','All file'}, 'Select image
file','/home/wangxw/DATA/others/figure.png');
if tag
    [imorig,map] = imread([impath,imfile]);
    if ~isempty(map)
        if any(map(1,:))
            imorig=ind2rgb(imorig,map);
        end
    end
    im =rgb2gray(imorig);
    imorig(1,1,3)=0;
    imshow(imorig);
    set(gcbo,'userdata',im);
    set(handles.fig,'userdata',imorig);
end
```

```
function roi1_Callback(hObject, eventdata, handles)
a=get(gcbo,'userdata');
if ~isempty(a)
    delete(a{1});
end
a=imrect;
set(gcbo,'userdata',{a});
rect=getPosition(a);
set(handles.firstroi,'string',mat2str(round(rect)));
addNewPositionCallback(a,@(p) set(handles.firstroi,'string',(mat2str(p,3))));
function firstroi_Callback(hObject, eventdata, handles)
function firstroi_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function roi2_Callback(hObject, eventdata, handles)
a=get(gcbo,'userdata');
if ~isempty(a)
    delete(a{1});
end
a=imrect;
rect=getPosition(a);
set(handles.firstroi,'string',mat2str(round(rect)));
set(gcbo,'userdata',{a});
addNewPositionCallback(a,@(p) set(handles.firstroi,'string',(mat2str(p,3))));
function secondroi_Callback(hObject, eventdata, handles)
function secondroi_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function ok_Callback(hObject, eventdata, handles)
a=get(handles.roi2,'userdata');
if isempty(a)
    disp('Error: -1');
else
    pos2=getPosition(a{1});
    a=get(handles.roi1,'userdata');
    pos1=getPosition(a{1});
    rstep=str2num(get(handles.secondroi,'string'));
    im=get(gcf,'userdata');
    imsize=size(im);
    step=abs((pos2(1:2)+pos2(3:4)/2-(pos1(1:2)+pos1(3:4)/2))./rstep);
    sz=pos2(3:4)/2+pos1(3:4)/2;
    startp=round(mod(pos2(1:2)+sz/2,step));
```

```
        row=round(startp(2):step(2):imsize(1)-sz(1));
        col=round(startp(1):step(1):imsize(2)-sz(2));
        block=zeros(length(row),length(col),5);
        block(:,:,3)=sz(1);
        block(:,:,4)=sz(2);
        for j=1:length(row)
            for i=1:length(col)
                block(j,i,1:2)=[col(i),row(j)];
            end
        end
        block(1,:,:)=[];block(:,1,:)=[];
        plotrectedge(block,im);
        set(handles.strect,'string',num2str(startp));
        set(handles.szrect,'string',num2str(round(step*10)/10));
        set(handles.rectsz,'string',num2str(round(sz)));
        set(gcbo,'userdata',block);
end
function shear_Callback(hObject, eventdata, handles)
im=get(gcf,'userdata');
[imt,sf]=imtran(im,1);
set(handles.accept,'userdata',imt);
set(handles.shearfactor,'string',num2str(sf));
function accept_Callback(hObject, eventdata, handles)
imt=get(handles.accept,'userdata');
set(gcf,'userdata',imt);
set(handles.imopen,'userdata',rgb2gray(imt));
function rot_Callback(hObject, eventdata, handles)
im=get(gcf,'userdata');
[imt,sf]=imtran(im,2);
set(handles.accept,'userdata',imt);
set(handles.shearfactor,'string',num2str(sf));
function test_Callback(hObject, eventdata, handles)
function evaluation_Callback(hObject, eventdata, handles)
im=get(gcf,'userdata');
block=get(handles.ok,'userdata');
plotrectedge(block,im);
function crop_Callback(hObject, eventdata, handles)
a=imcrop;
a=double(a(:,:,1));
hist(a(:),25);
function reimage_Callback(hObject, eventdata, handles)
a=get(gcf,'userdata');
imshow(a);
function strect_Callback(hObject, eventdata, handles)
```

```
function strect_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function szrect_Callback(hObject, eventdata, handles)
function szrect_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function setmesh_Callback(hObject, eventdata, handles)
im=get(gcf,'userdata');
block=get(handles.ok,'userdata');
imsize=size(im);
startp=str2num(get(handles.strect,'string'));
step=str2num(get(handles.szrect,'string'));
sz=str2num(get(handles.rectsz,'string'));
    row=round(startp(2):step(2):imsize(1)-sz(1));
    col=round(startp(1):step(1):imsize(2)-sz(2));
    block=zeros(length(row),length(col),5);
    block(:,:,3)=sz(1);
    block(:,:,4)=sz(2);
    for j=1:length(row)
        for i=1:length(col)
            block(j,i,1:2)=[col(i),row(j)];
        end
    end
    plotrectedge(block,im);
    set(handles.ok,'userdata',block);
function enhance_Callback(hObject, eventdata, handles)
im=get(gcf,'userdata');
BW = edge(im(:,:,1),'canny');
BW_filled = imfill(BW,'holes');
imshow(BW_filled);
function region_Callback(hObject, eventdata, handles)
imorig=get(gcf,'userdata');
im=get(handles.imopen,'userdata');
block=get(handles.ok,'userdata');
szblock=size(block);
for i=1:szblock(1)
    for j=1:szblock(2)
        curblock=round(block(i,j,:));
        seed=[min(curblock(3),curblock(1)-1),min(curblock(4),curblock(2)-1)];
        subpos_lt=[curblock(1)-seed(1),curblock(2)-seed(2),2*curblock(3)+1,2*curblock(4)+1];
```

```
subim=im(subpos_lt(2):subpos_lt(2)+subpos_lt(4)-1,subpos_lt(1):subpos_lt(1)+subpos_lt(3)-1);
        th=1;
        tag=0;
        blockim_old=subim<0;
        n=20;
        while th<=n && tag==0
            blockim=region(subim,seed,th);
            area=sum(blockim(:));
            [y,x]=find(blockim);
            sz=[max(x)-min(x),max(y)-min(y)];
            if area>=subpos_lt(3)*subpos_lt(4)/2 || any((sz-0.7* subpos_lt(3:4)) >0)
                tag=1;
                if th>2.5
                    blockim=blockim_old;
                    [y,x]=find(blockim);
                    sz=[max(x)-min(x),max(y)-min(y)];
                    if all(sz>4)

block(i,j,:)=[subpos_lt(1)-1+(max(x)+min(x))/2,subpos_lt(2)-1+(max(y)+min(y))/2,sz,2];
                    else
                        block(i,j,end)=-1;
                    end
                else
                    block(i,j,end)=-1;
                end
            else
                blockim_old=blockim;
                th=th+1;
            end
        end
        if tag==0
            block(i,j,end)=-1;
        end
        if th>=n && area>=subpos_lt(3)*subpos_lt(4)/16 && all(sz>4)

block(i,j,:)=[subpos_lt(1)-1+(max(x)+min(x))/2,subpos_lt(2)-1+(max(y)+min(y))/2,sz,3];
        end
    end
end
plotrectedge(block,imorig);
set(handles.ok,'userdata',block);
function th_Callback(hObject, eventdata, handles)
imorig=get(gcf,'userdata');
im=get(handles.imopen,'userdata');
```

```
block=get(handles.ok,'userdata');
block=thresh(block,im);
plotrectedge(block,imorig);
set(handles.ok,'userdata',block);
function recog_Callback(hObject, eventdata, handles)
function handy_Callback(hObject, eventdata, handles)
block=get(handles.ok,'userdata');
hrect=imrect;
rect=wait(hrect);
[i,j]=findblock(block,rect);
block(j,i,:)=[rect(1)+rect(3)/2,rect(2)+rect(4)/2,rect(3:4),3];
set(handles.ok,'userdata',block);
im=get(gcf,'userdata');
plotrectedge(block,im);
function correct_Callback(hObject, eventdata, handles)
block=get(handles.ok,'userdata');
for i=1:size(block,1)
    rc=mean(block(i,block(i,:,5)>1.5,1:4),2);
    if ~any(isnan(rc)) && all(rc>4)
    for j=1:size(block,2)
        if block(i,j,5)<=0 || abs(block(i,j,2)-rc(2))>rc(4)/4 || abs(block(i,j,4)-rc(4))>rc(4)/3
            block(i,j,2)=rc(2);
            block(i,j,4:5)=[rc(4),1];
        end
    end
    end
end
for j=1:size(block,2)
    rc=mean(block(block(:,j,5)>1.5,j,1:4),1);
    if ~any(isnan(rc)) && all(rc>4)
    for i=1:size(block,1)
        if block(i,j,5)<=0 || abs(block(i,j,1)-rc(1))>rc(3)/4 || abs(block(i,j,3)-rc(3))>rc(3)/3
            block(i,j,1)=rc(1);
            block(i,j,3)=rc(3);
            block(i,j,5)=1;
        end
    end
    end
end
im=get(gcf,'userdata');
plotrectedge(block,im);
set(handles.ok,'userdata',block);
function menubar_Callback(hObject, eventdata, handles)
set(gcf,'menubar','default','resize','on');
```

```
function bg_Callback(hObject, eventdata, handles)
im=get(gcf,'userdata');
block=get(handles.ok,'userdata');
im=localbg(im,block);
im=plotrectedge(block,im);
function rectsz_Callback(hObject, eventdata, handles)
function rectsz_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function savedata_Callback(hObject, eventdata, handles)
block=get(handles.ok,'userdata');
savedata(handles,block);
```

(2) preview.m

```
function res=preview(im,block,row,col,me)
sz=size(block);
res=zeros(sz(1),sz(2));
res1=res;
for i=row
    for j=col
        curblock=ceil(block(i,j,:));
        rowst=ceil(curblock(2)-curblock(4)/2);
        colst=ceil(curblock(1)-curblock(3)/2);
        subim=double(im(rowst:rowst+curblock(4)-1,colst:colst+curblock(3)-1));
        switch me
            case 3
                res(i,j)=max(subim(:));
                res1(i,j)=min(subim(:));
            case 2
                res(i,j)=median(subim(:));
            otherwise
                res(i,j)=mean2(subim);
        end
    end
end
if me<3
    res1=res;
end
res(:,:,2)=res1;
```

(3) imtran.m

```
function [imt,shearfactor]=imtran(im,hvr)
imshow(im);
[x,y]=ginput(2);
imsz=size(im);
dx=x(1)-x(2);
dy=y(1)-y(2);
trfm=diag(ones(3,1));
if abs(dx)<abs(dy)
        disp('shear in vertical');
        shearfactor=-1*sign(dy)*dx/imsz(1);
        trfm(2,1)=shearfactor;
else
        disp('shear affine in horizental');
        shearfactor=-1*sign(dx)*dy/imsz(2);
        trfm(1,2)=shearfactor;
end
if hvr==2
        disp('              Or    Rotated');
      q=atan(shearfactor);
      cosq=cos(q);
      sinq=sin(q);
      trfm=[cosq sinq 0;-sinq cosq 0 ; 0 0 1];
end
imtrf=maketform('affine',trfm);
[imt,x,y]=imtransform(im,imtrf);
imshow(imt);
```

(4) mkrect.m

```
function mkrect
  [imfile, impath] = uigetfile({'*.tif','TIFF file';'*.*','All file'}, 'Select image file');
imorig = imread([impath,imfile]);
imshow(imorig);
im =rgb2gray(imorig);
operation = questdlg('Set the position and size?');
switch operation
        case 'OK'
        case 'Cancel'
        otherwise
end
imshow(imorig);
a=imrect;[560 535.8125 16 14.375]
```

(5) localbg.m

```
function im=localbg(im,block)
for i=1:size(block,1)
    for j=1:size(block,2)
        curblock=block(i,j,:);
        seed=[min(curblock(3),curblock(1)-1),min(curblock(4),curblock(2)-1)];
        subpos_lt=[curblock(1)-seed(1),curblock(2)-seed(2),2*curblock(3)+1,2*curblock(4)+1];
        pattern=zeros( subpos_lt(4), subpos_lt(3));
        top=pattern;top(1:ceil(curblock(4)/4),:)=1;
        botom=pattern;botom(ceil(curblock(4)*3.5/2):end,:)=1;
        left=pattern;left(:,1:ceil(curblock(3)/4))=1;
        right=pattern;right(:,ceil(curblock(3)*3.5/2):end,:)=1;
        pattern=(top+botom+left+right)>0.5;
        im( subpos_lt(2): subpos_lt(2)+ subpos_lt(4)-1,subpos_lt(1): subpos_lt(1)+
subpos_lt(3)-1,3)=pattern*255;
    end
end
```

(6) findblock.m

```
function [col,row]=findblock(block,seed)
cx=block(1,:,1);
cy=block(:,1,2);
x=cx-seed(1);
i=find(x>0,1,'first');
y=cy-seed(2);
j=find(y>0,1,'first');
subblock=block(j-1:j+1,i-1:i+1,1:2);
seeds=(seed(1)+seed(3)/2)*ones(3,3);
seeds(:,:,2)=(seed(2)+seed(4)/2)*ones(3,3);
dis=sum((subblock-seeds).^2,3);
[k,l]=find(dis==min(dis(:)));
col=i-2+l;
row=j-2+k;
```

(7) plotrectedge.m

```
function im=plotrectedge(block,im,haxes,cand)
if nargin<2
    im=uint8(zeros(block(end,end,2)+block(end,end,4),block(end,end,3)+block(end,end,1),3));
end
imsz=size(im);
blocksz=size(block);
imblock=zeros(imsz(1),imsz(2));
for i=1:blocksz(1)
```

```
        for j=1:blocksz(2)
            curblock=ceil(block(i,j,:));
            blockedge=50*(curblock(5)+2)*ones(curblock(4),curblock(3));
            blockcenter=zeros(curblock(4)-2,curblock(3)-2);
            blockedge(2:end-1,2:end-1)=blockcenter;
            rowst=ceil(curblock(2)-curblock(4)/2);
            colst=ceil(curblock(1)-curblock(3)/2);
            imblock(rowst:rowst+curblock(4)-1,colst:colst+curblock(3)-1)=blockedge;
        end
    end
    im(:,:,2)=uint8(double(im(:,:,2))+imblock);
    if size(im,3)<3
        im(:,:,3)=0;
    end
    if nargin>=3
        cand=round(cand);
        if cand(1)==2
            im(:,cand(2):cand(3),3)=255;
            im(:,cand(4),2:3)=255;
        else
            im(cand(2):cand(3),:,3)=255;
            im(cand(4),:,2:3)=255;
        end
        imshow(im,'parent',haxes);
    else
        imshow(im);
    end
```

(8) region.m

```
function imnew=region(im,seed,th)
  [row,col]=size(im);
imnew=zeros(row,col);
numRegion=0;
for num=1:size(seed,1)
    ypos=seed(num,1);
    xpos=seed(num,2);
    if imnew(ypos,xpos)==0
        numRegion=numRegion+1;
        queue=[ypos,xpos];
        tag=1;
        while tag~=0
            row_buttom=queue(1,1);
            col_buttom=queue(1,2);
```

```
                pixel_buttom=im(row_buttom,col_buttom);
                for i=-1:1
                    for j=-1:1
                        if
row_buttom+i<=row&row_buttom+i>0&col_buttom+j<=col&col_buttom+j>0
                            if
imnew(row_buttom+i,col_buttom+j)~=numRegion&abs(double(im(row_buttom+i,col_buttom+j))-dou
ble(pixel_buttom))<=th
                                tag=tag+1;
                                queue(tag,:)=[row_buttom+i,col_buttom+j];
                                imnew(row_buttom+i,col_buttom+j)=numRegion;
                            end
                        end
                    end
                end
                queue=queue(2:tag,:);
                tag=tag-1;
            end
        end
end
```

(9) thresh.m

```
function blknew=thresh(block,im)
if size(im,3)>1.5
    im=rgb2gray(im);
end
blknew=block;
imsize=size(im);
blksize=size(block);
blkwh=round(1.5*[max(median(block(:,:,3))),max(median(block(:,:,4)))]);
for i=1:blksize(1)
    for j=1:blksize(2)
        curblk=block(i,j,:);

rectold=[max(curblk(2)-blkwh(2),1),max(curblk(1)-blkwh(1),1),min(curblk(2)+blkwh(2),imsize(1)),mi
n(curblk(1)+blkwh(1),imsize(2))];
        subim=im(rectold(1):rectold(3),rectold(2):rectold(4));
        rect=threeLevel(subim);
        area=abs(blkwh(1)*blkwh(2)-rect(3)*rect(4))/(blkwh(1)*blkwh(2));
        width=abs(blkwh(2)-rect(3))/blkwh(2);
        height=abs(blkwh(1)-rect(4))/blkwh(1);
        blktag= area>2 || width>0.4 || height> 0.4;
        if ~blktag
            blknew(i,j,:)=[rectold(2)+rect(2),rectold(1)+rect(1),rect(4),rect(3),1];
```

```
            end
        end
end
function rect=threeLevel(subim)
im=(subim(3:end,2:end-1)+subim(2:end-1,2:end-1)+subim(1:end-2,2:end-1)+subim(2:end-1,1:end-2)+
subim(2:end-1,3:end))/5;
imsize=size(im);
ims=round(imsize/2);
immax=max(max(im(ims(1)-5:ims(1)+5,ims(2)-5:ims(2)+5)));
immin=min(im(:));
th=immax-immin;
im2=im>th*2/3;
im3=bwselect(im2,round(imsize/2),round(imsize/2),8);
 [r,c]=find(im3);
if length(r)>2
        rect1=[min(r),min(c)];
        rect2=[max(r),max(c)];
        col=sum(im3,1);
        row=sum(im3,2);
        wh=rect2-rect1+1;
        im3(:,(col)< 0.33*wh(2))=0;
        im3((row)< 0.33*wh(1),:)=0;
        [r1,c1]=find(im3);
        if length(r1)>2
            rect3=[min(r1),min(c1)];
            rect4=[max(r1),max(c1)];
            rect=[ceil((rect3+rect4)/2),rect4-rect3+1];
        else
            rect=[ceil((rect1+rect2)/2),wh];
        end
else
        rect=[0 0 0 0];
end
```

(10) savedata.m

```
function varargout = savedata(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',             mfilename, ...
                   'gui_Singleton',    gui_Singleton, ...
                   'gui_OpeningFcn', @savedata_OpeningFcn, ...
                   'gui_OutputFcn',    @savedata_OutputFcn, ...
                   'gui_LayoutFcn',    [] , ...
                   'gui_Callback',      []);
if nargin && ischar(varargin{1})
```

```
        gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
        gui_mainfcn(gui_State, varargin{:});
end
function savedata_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
block= varargin{2};
handles.block= block;
handles.hparent = varargin{1};
sz = size(block);
str=num2str((1:sz(1))');
set(handles.first,'string',str);
set(handles.last,'string',str);
set(handles.curr,'string',str);
set(handles.rc,'userdata',sz);
guidata(hObject, handles);
function varargout = savedata_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function spath_Callback(hObject, eventdata, handles)
function spath_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
end
function browser_Callback(hObject, eventdata, handles)
  [fn,fp,tag]=uiputfile({'*.txt';'*.jpg'},'Save result','output');
if tag~=0
        set(handles.spath,'string',[fp,fn]);
end
function export_Callback(hObject, eventdata, handles)
fn=get(handles.spath,'string');
fmt=fn(end-2:end);
if strcmp(fmt,'jpg')
        set(gcf,'PaperPositionMode','auto')
        print('-djpeg','-r200',fn);
elseif strcmp(fmt,'txt')
        me=get(handles.mmm,'value');
        rc=get(handles.rc,'value');
        block=handles.block;
        sz=size(block);
        imgray=get(handles.hparent.imopen,'userdata');
        res1=get(handles.Preview,'userdata');
```

```
        fid =fopen(fn,'w+');
        if rc==2
            fprintf(fid,'Col\r\n');
            row=1:sz(1);
            fid2=fopen('tmp238979','w+');
            for j=1:sz(2)
                res=preview(imgray,block,row,j,me);
                fprintf(fid2,'%12.8f\t',res(:,j,1)-res1);
                fprintf(fid2,'\r\n');
            end
            fclose(fid2);
            data=load('tmp238979');
            for i=1:sz(1)

                fprintf(fid,'%12.8f\t',data(:,i));
                fprintf(fid,'\r\n');
            end
            !rm 'tmp238979'
        else
            fprintf(fid,'Row\r\n');
            col=1:sz(2);
            for j=1:sz(1)
                res=preview(imgray,block,j,col,me);
                fprintf(fid,'%g\t',res(j,:,1)-res1);
                fprintf(fid,'\r\n');
            end
        end
        fclose(fid);
else
        disp('Error -1: non-surpporting format');
end
function last_Callback(hObject, eventdata, handles)
function last_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
end
function first_Callback(hObject, eventdata, handles)
cur=get(gcbo,'value');
str=get(gcbo,'string');
set(handles.last,'string',str(cur:end,:));
set(handles.last,'value',1);
function first_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
```

```
end
function Preview_Callback(hObject, eventdata, handles)
cf=get(handles.first,'value');
cl=get(handles.last,'value')+cf-1;
me=get(handles.mmm,'value');
rc=get(handles.rc,'value');
curr=get(handles.curr,'value');
block=handles.block;
im=get(handles.hparent.fig,'userdata');
imgray=get(handles.hparent.imopen,'userdata');
if rc==2
    cand=[rc,block(1,cf,1),block(1,cl,1),block(1,curr,1)];
else
    cand=[rc,block(cf,1,2),block(cl,1,2),block(curr,1,2)];
end
plotrectedge(block,im,handles.hparent.ax,cand);
sz=size(block);
if rc==2
    row=1:sz(1);
    col=curr;
    res=preview(imgray,block,row,col,me);
    res=res(:,col,1);
    res1=zeros(sz(1),1);
    for i=cf:cl
        res2=preview(imgray,block,row,i,me);
        res1(:,i-cf+1)=res2(:,i,2);
    end
    res1=mean(res1,2);
else
    row=curr;
    col=1:sz(2);
    res=preview(imgray,block,row,col,me);
    res=res(row,:,1);
    res1=zeros(1,sz(2));
    for i=cf:cl
        res2=preview(imgray,block,i,col,me);
        res1(i-cf+1,:)=res2(i,:,2);
    end
    res1=mean(res1,1);
end
res=res-res1;
h=plot(res);
ylim=get(gca,'ylim');
set(h,'linestyle','none','marker','*','markersize',8,'color','r');
```

```
set(gca,'ylim',ylim.*[0.75 1.25]);
set(gcbo,'userdata',res1);
function curr_Callback(hObject, eventdata, handles)
function curr_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function rc_Callback(hObject, eventdata, handles)
cur=get(gcbo,'value');
blocksz=get(gcbo,'userdata');
if cur==2
    str=num2str((1:blocksz(2))');
    set(handles.first,'string',str);
    set(handles.last,'string',str);
    set(handles.curr,'string',str);
else
    str=num2str((1:blocksz(1))');
    set(handles.first,'string',str);
    set(handles.first,'string',str);
    set(handles.curr,'string',str);
end
function rc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function mmm_Callback(hObject, eventdata, handles)
function mmm_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Table S1 The output data of HIV immunoassay (30 sera) [a]

	gp41 [b]	gp36	gp120	p55	p51	gp41	p31	p24	5% BSA
Serum No.30	8.66863	6.25589	1.48667	7.62015	7.35578	12.7375	13.4037	9.64283	-0.322715
Serum No.29	5.79103	4.59418	5.74377	3.86859	4.72161	8.56925	6.94321	13.647	0.209314
Serum No.28	7.06804	5.45724	6.95516	2.51316	10.1695	12.7164	11.4027	8.87344	1.0651
Serum No.27	9.31267	2.76593	0.672957	6.80921	10.5272	13.3991	12.3276	13.7763	-0.27268
Serum No.26	11.1342	4.53324	1.33899	24.0414	11.057	13.206	16.3914	13.8478	0.65097
Serum No.25	7.86496	8.42884	7.22161	12.1394	12.4725	10.6551	16.856	18.0651	2.70447
Serum No.24	12.7581	4.04346	7.23148	10.4378	10.1515	14.4623	18.3347	13.2379	0.669321
Serum No.23	11.5843	8.30177	4.73632	3.68144	6.36998	14.1188	14.7434	14.3411	0.0934903
Serum No.22	14.3573	8.07289	7.0367	11.2064	10.724	16.9321	18.5641	26.4429	3.97663
Serum No.21	11.6802	13.4126	1.33397	2.76697	14.2019	16.531	21.5054	12.6806	1.69027
Serum No.20	7.88158	14.6411	10.2265	13.2291	14.4981	19.5379	19.9938	23.2495	2.98528
Serum No.19	6.67278	10.9977	10.642	9.74256	10.5731	12.5862	13.1434	15.2381	0.697022
Serum No.18	-1.00242	2.76541	4.29536	2.05817	4.32808	2.77337	1.35145	4.74169	-0.384868
Serum No.17	5.18542	11.7692	11.6634	10.3099	8.42348	10.2225	13.5783	20.5677	-0.600242
Serum No.16	3.8518	11.2076	11.656	4.08328	2.82635	7.57497	11.0544	13.813	-0.671572
Serum No.15	3.60942	8.56544	7.33431	3.00883	1.40824	5.89024	7.66603	11.3234	-0.486323
Serum No.14	6.41257	10.9333	9.7294	6.47524	4.12933	9.64578	12.2822	15.3944	-0.331025
Serum No.13	2.17936	8.47039	9.66066	1.7893	3.12621	5.75121	7.06596	13.8116	-0.502251
Serum No.12	3.19287	6.82912	8.7365	3.57739	2.32081	7.65651	9.41759	10.7405	-0.553151
Serum No.11	1.93854	5.92365	6.3028	1.27528	2.63504	4.52181	11.9517	8.17763	-0.321503
Serum No.10	6.22576	10.9531	13.2827	10.5483	4.84539	11.4913	13.0833	25.7973	2.13712
Serum No.9	4.89491	5.80661	4.37413	2.96139	1.82739	8.79294	4.3028	8.45481	-0.460873
Serum No.8	5.44131	7.39145	10.844	11.7964	3.9839	9.81908	12.8317	22.9659	-0.333622
Serum No.7	10.3712	6.97507	8.97576	11.0407	4.07912	8.3795	16.2038	34.3115	2.21312
Serum No.6	9.43057	7.5651	17.4969	18.4619	10.6468	15.6399	20.5473	33.5871	4.06094
Serum No.5	9.26264	10.3315	18.5383	19.5156	7.90235	12.6247	19.4536	42.0303	1.50416
Serum No.4	4.8563	3.59418	12.7071	18.0905	5.22057	6.71382	16.3257	46.7381	2.15045

Serum No.3	10.0609	6.68871	11.0346	14.5822	5.71762	10.6596	20.6704	38.0073	3.09124
Serum No.2	7.9349	8.55661	12.879	16.5566	8.34089	9.26575	21.597	28.6089	1.88158
Serum No.1	10.242	10.7043	16.0274	16.5518	10.339	15.6868	20.6427	48.9675	3.759
5% BSA	1.42244	0.601454	0.313366	1.4955	2.12448	3.18906	2.27337	8.01714	1.43837
HIV$^-$	0	0	0	0	0	0	0	0	0
HIV$^+$	10.213	5.88937	10.281	6.19391	10.7341	12.6037	12.6887	11.117	0.379155

[a] All values are the median (median is more robust in the presence of outlier values than mean.) of relative intensity of each sport showed in Fig. 3A.

[b] This column was used as the boundary indicator, so it will be not illustrated into Fig. 3B.