External modelling S2

Infrared based saliva screening test for COVID-19

Authors: Bayden R. Wood(1)*†, Kamila Kochan(1)†, Diana E. Bedolla(1,2,3)†, Natalia Salazar-Quiroz(4), Samantha Grimley(4), David Perez-Guaita(1,5), Matthew J. Baker(6), Jitraporn Vongsvivut(7), Mark Tobin(7), Keith Bambery(7), Dale Christensen(7), Shivani Pasricha(4), Anthony K. Eden(1), Aaron Mclean(1), Supti Roy(1), Jason Roberts(8), Julian Druce(8), Deborah A. Williamson(4,9), Julie McAuley(4), Mike Catton(8), Damian Purcell(4)*, Dale Godfrey(4)*, and Philip Heraud(1,10) *

Affiliations:

1. Centre for Biospectroscopy and School of Chemistry, Monash University, Clayton Campus, 3800, Victoria, Australia.
2. Elettra−Sincrotrone Trieste, S.S. 14 Km 163.5, 34149, Trieste, Italy.
3. Area Science Park, Padriciano 99, Padriciano, TS, 34149, Italy.
4. Department of Microbiology and Immunology, University of Melbourne, Melbourne, Australia.
5. FOCAS Research Institute, Dublin Institute of Technology, D04, Dublin, Ireland.
6. Department of Pure and Applied Chemistry University of Strathclyde, Technology and Innovation Centre Level 7, 99 George Street, Glasgow, G1 1RD, Scotland, UK.
7. Australian Synchrotron, Victoria 3168, Australia.
8. Victorian Infectious Diseases Reference Laboratory, The Peter Doherty Institute for Infection and Immunity, Melbourne, Vic, Australia;
9. Department of Microbiology, Royal Melbourne Hospital, Melbourne, Australia.
10. Department of Microbiology and the Biomedicine Discovery Institute, Faculty of Medicine, Nursing and Health Sciences, Monash University, Victoria, Australia.

†These authors contributed equally to this work. *Correspondence to: bayden.wood@monash.edu (B.R.W.); phil.heraud@monash.edu (P.H.); dfjp@unimelb.edu.au (D.P.); godfrey@unimelb.edu.au (D.G.)

Monte Carlo Double Cross validation modelling of transflection infrared spectra of patient saliva samples

Table of Contents

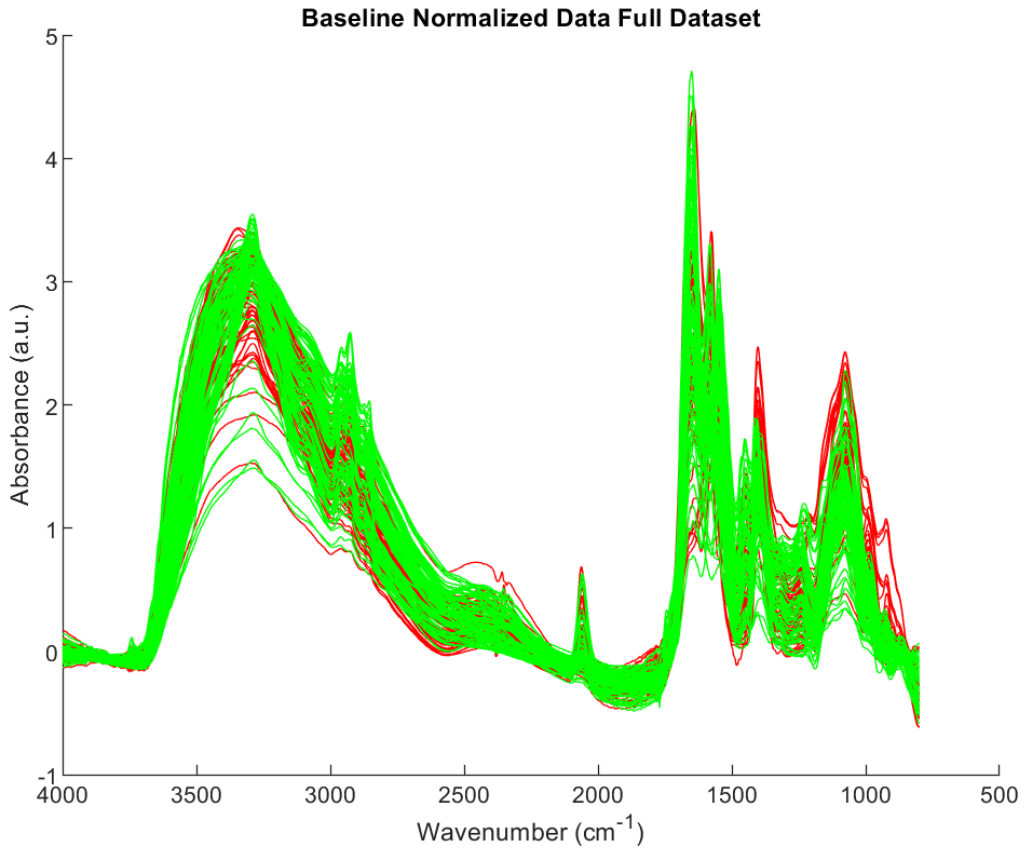
1. Load Whole Dataset

```
clear
close

addpath(genpath('Optimodel_Functions'))
[num,txt]=xlsread('Dataset.xlsx');
WDATA=dataset(num(2:end,:));
WDATA.axisscale{2,1}=num(1,:);
WDATA.label{1,1}=txt(2:end,1);
WDATA.class{1,1}=txt(2:end,2);
WDATA.class{1,2}=txt(2:end,3);

WDATAN=preprocess('calibrate',...
    [preprocess('default','SNV') preprocess('default','baseline')],WDATA);

figure
plot(WDATAN.axisscale{2,1},WDATAN.data(WDATAN.class{1,1}==1,:),'r')
hold on
plot(WDATAN.axisscale{2,1},WDATAN.data(WDATAN.class{1,1}==2,:),'g')
ax=gca;ax.FontSize=8;
ax.XLabel.String='Wavenumber (cm^-^1)'; ax.XLabel.FontSize=8;ax.XDir='reverse';
ax.YLabel.String='Absorbance (a.u.)';ax.YLabel.FontSize=8;
box off
title('Baseline Normalized Data Full Dataset');
```




```
tabulate(WDATA.classid{1,1})
```

```
    Value     Count    Percent
    NegSal       84     49.12%
    PosSal       87     50.88%
```

2. Define Preprocessing and Spectral Regions

```
MC=preprocess('default','Mean Center');
FD=preprocess('default','Derivative (SavGol)');
FD.userdata.useexcluded ='true';
FD.userdata.wt ='1/d';
FD.userdata.width=11;
SNV=preprocess('default','SNV');
prepo= [FD SNV MC];
reg=[800 1300];
```

3. Do Double Cross Validation

It is given the option to perform the MCCV or to use previous resoulds saved as "MCCV_out"

```
Answer=inputdlg('Do Model=1 / Load Model output=2');

if Answer{1,1}==num2str(1)
[ERR_PLS,ERR_SVM,CVerr,LVsel]=MC2CV(WDATA,1,WDATA.class{1,2},50,0.7,8,prepo,reg);
elseif Answer{1,1}==num2str(2)
load MCCV_out
end
```

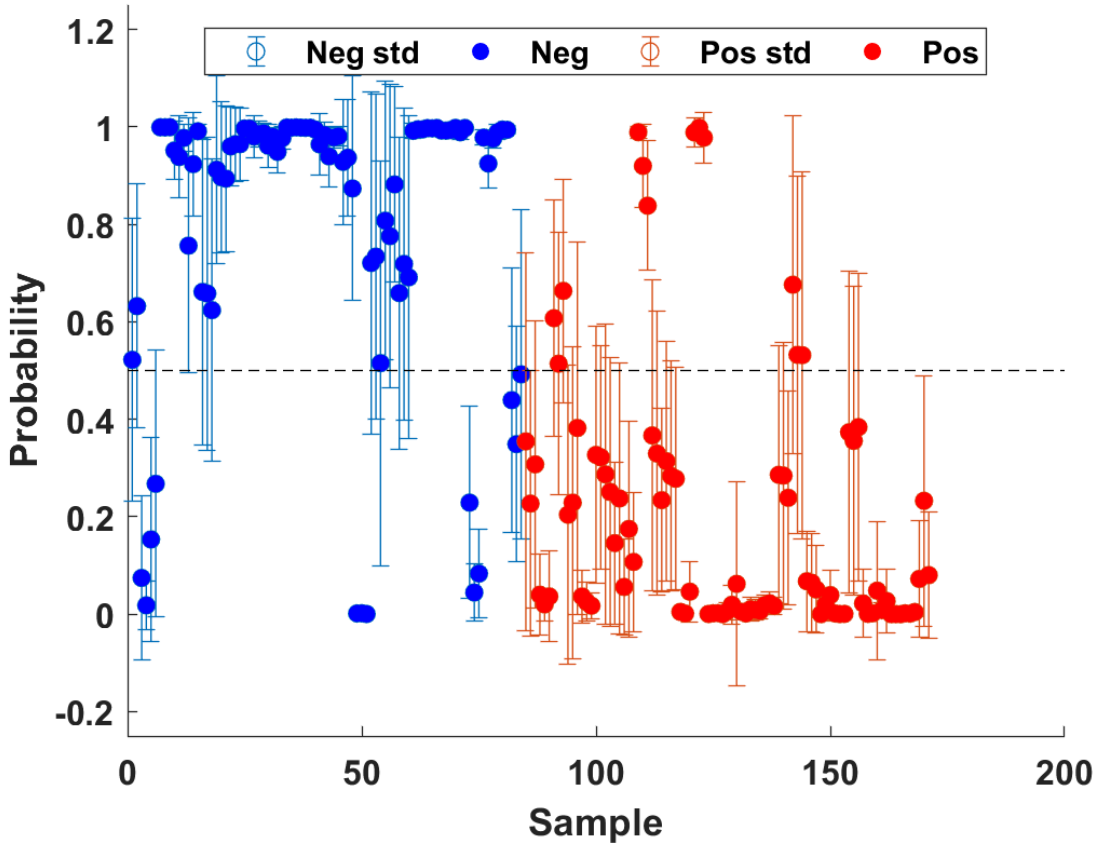
4. Plot Results

4.1 PLSDA- 3 Replicates

```
 Avprob_pls=nanmean(ERR_PLS.Yprob_MCCV');
 Stdprob_pls=nanstd(ERR_PLS.Yprob_MCCV');
```

Plot Probability

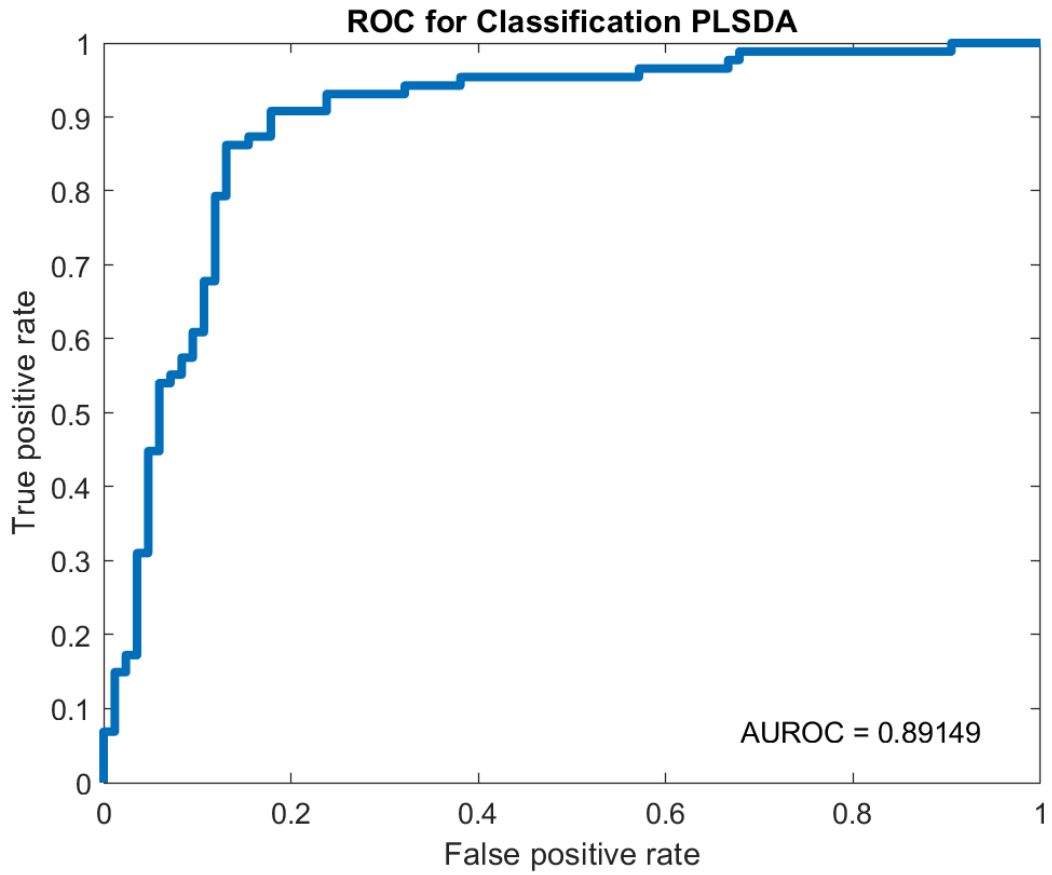
```
figure
negind=find((WDATA.class{1,1}==1));
set(gca,'Ylim',[-0.25 1.25], 'FontSize',12,'FontName','Helvetica',...
    'FontWeight', 'bold', 'Box', 'Off')
hold on
errorbar(negind,Avprob_pls(negind),Stdprob_pls(negind),'o')
scatter(negind,Avprob_pls(negind),'b','filled')
hold on
posind=find((WDATA.class{1,1}==2));
errorbar(posind,Avprob_pls(posind),Stdprob_pls(posind),'o')
scatter(posind,Avprob_pls(posind),'r','filled')
hline(0.5,'--k')
lgd=legend('Neg std','Neg','Pos std','Pos');
lgd.NumColumns = 4;
lgd.Location = 'north';
ax=gca; ax.YLabel.String='Probability';ax.XLabel.String='Sample';
```



Plot ROC

```
[Xpls,Ypls,T_PLS,AUC_PLS_CV] = perfcurve(WDATA.class{1,1},1-Avprob_pls,2);

ROC=figure;
plot(Xpls,Ypls,"LineWidth",3)
xlabel('False positive rate')
ylabel('True positive rate')
title(['ROC for Classification PLSDA'])
text(0.68, 0.070,['AUROC = ' num2str(AUC_PLS_CV)])
```



4.2 PLSDA- Average

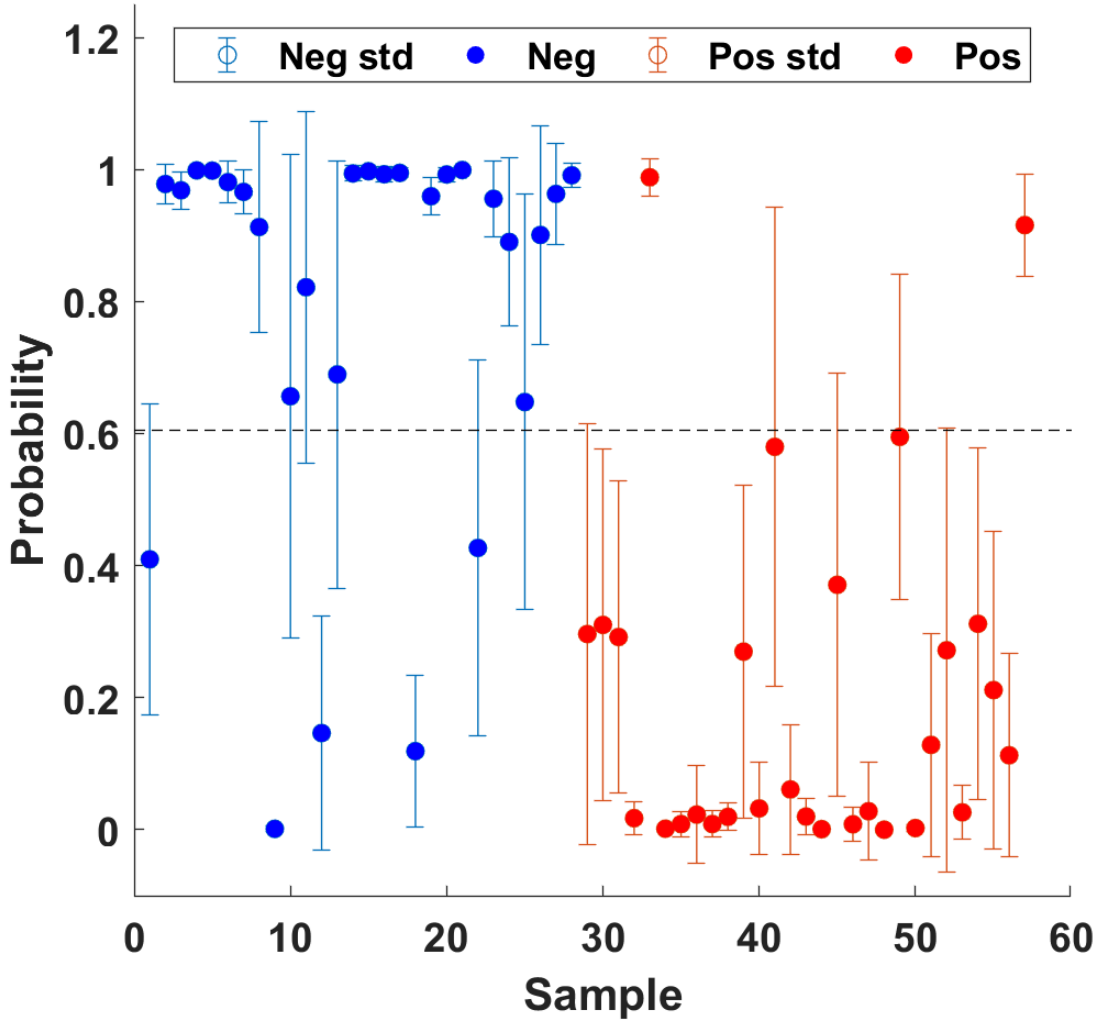
```
dat_Avprob_pls=dataset(Avprob_pls');
dat_Avprob_pls.class{1,1}=WDATA.class{1,1};
dat_Avprob_pls.class{1,2}=WDATA.class{1,2};
rep_Avprob_pls=replicateaverager(dat_Avprob_pls,2);


dat_Stdprob_pls=dataset(Stdprob_pls');
dat_Stdprob_pls.class{1,1}=WDATA.class{1,1};
dat_Stdprob_pls.class{1,2}=WDATA.class{1,2};
rep_Stdprob_pls=replicateaverager(dat_Stdprob_pls,2);
```

Plot Probablity

```
figure
set(gcf, 'Position',[1000 200 550 500], 'PaperPositionMode', 'Auto');
negind=find((rep_Avprob_pls.class{1,1}==1));
hold on
set(gca,'Ylim',[-0.1 1.25], 'FontSize',14,'FontName','Helvetica', 'FontWeight', 'bold',...
    'Box', 'Off')
errorbar(negind,rep_Avprob_pls.data(negind),rep_Stdprob_pls.data(negind),'o')
```

```
scatter(negind,rep_Avprob_pls.data(negind),'b','filled')
hold on
posind=find((rep_Avprob_pls.class{1,1}==2));
errorbar(posind,rep_Avprob_pls.data(posind),rep_Stdprob_pls.data(posind),'o')
scatter(posind,rep_Avprob_pls.data(posind),'r','filled')
hline(0.605,'--k')
lgd=legend('Neg std','Neg','Pos std','Pos');
lgd.NumColumns = 4;
lgd.Location = 'north';
ax=gca; ax.YLabel.String='Probability';ax.XLabel.String='Sample';
```



Plot ROC

```
[Xpls,Ypls,T_PLS,AUC_PLS_CV_rep] = perfcurve(rep_Avprob_pls.class{1,1},1-rep_Avprob_pls.data,2)
ROC_mat_PLS = [1-Xpls, Xpls, Ypls, T_PLS, 1-T_PLS];

ROC=figure;
set(gcf, 'Position',[1000 200 500 500], 'PaperPositionMode', 'Auto');
set(gca, 'FontSize',14,'FontName','Helvetica', 'FontWeight', 'bold', 'Box', 'Off')
```

```
plot(Xpls,Ypls,"LineWidth",3)
hold on,
plot(Xpls(33), Ypls(33), 'o', 'LineWidth', 3)
xlabel('False positive rate')
ylabel('True positive rate')
xticks([0:0.1:1])
text(0.68, 0.070,['AUROC = ' num2str(AUC_PLS_CV_rep)])
```