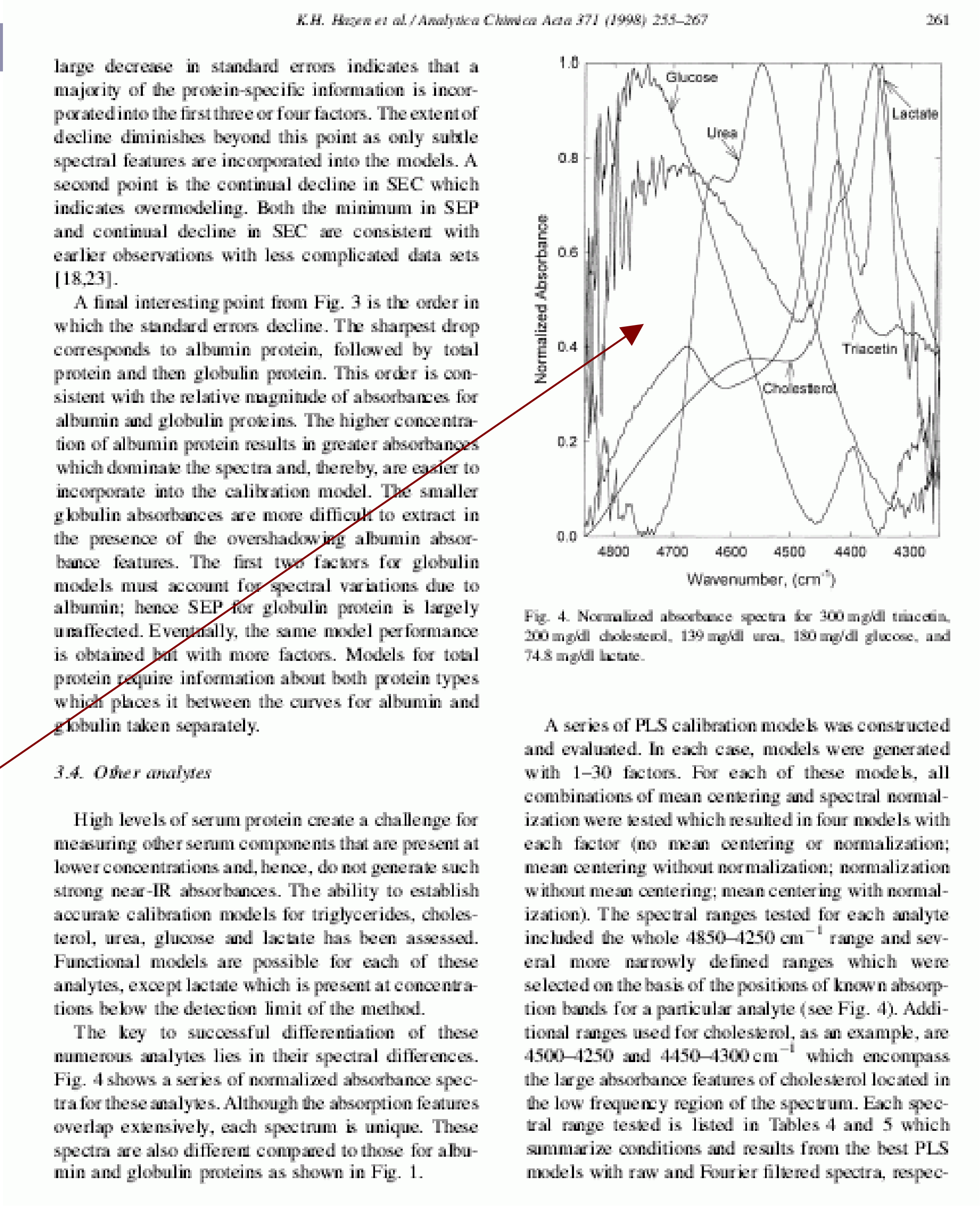
Reverse Engineering of Scientific Papers – Get Data out of *.pdf's with Perl

Andy Adler

Just because you're not paranoid
It doesn't mean they're not out to get you

Problem

- You need this data, but you only have the pdf



K.H. Hazen et al. / Analytica Chimica Acta 371 (1998) 255–267 261

large decrease in standard errors indicates that a majority of the protein-specific information is incorporated into the first three or four factors. The extent of decline diminishes beyond this point as only subtle spectral features are incorporated into the models. A second point is the continual decline in SEC which indicates overmodeling. Both the minimum in SEP and continual decline in SEC are consistent with earlier observations with less complicated data sets [18,23].

A final interesting point from Fig. 3 is the order in which the standard errors decline. The sharpest drop corresponds to albumin protein, followed by total protein and then globulin protein. This order is consistent with the relative magnitude of absorbances for albumin and globulin proteins. The higher concentration of albumin protein results in greater absorbances which dominate the spectra and, thereby, are easier to incorporate into the calibration model. The smaller globulin absorbances are more difficult to extract in the presence of the overshadowing albumin absorbance features. The first two factors for globulin models must account for spectral variations due to albumin; hence SEP for globulin protein is largely unaffected. Eventually, the same model performance is obtained but with more factors. Models for total protein require information about both protein types which places it between the curves for albumin and globulin taken separately.

3.4. Other analytes

High levels of serum protein create a challenge for measuring other serum components that are present at lower concentrations and, hence, do not generate such strong near-IR absorbances. The ability to establish accurate calibration models for triglycerides, cholesterol, urea, glucose and lactate has been assessed. Functional models are possible for each of these analytes, except lactate which is present at concentrations below the detection limit of the method.

The key to successful differentiation of these numerous analytes lies in their spectral differences. Fig. 4 shows a series of normalized absorbance spectra for these analytes. Although the absorption features overlap extensively, each spectrum is unique. These spectra are also different compared to those for albumin and globulin proteins as shown in Fig. 1.

Fig. 4. Normalized absorbance spectra for 300 mg/dl triacetin, 200 mg/dl cholesterol, 139 mg/dl urea, 180 mg/dl glucose, and 74.8 mg/dl lactate.

A series of PLS calibration models was constructed and evaluated. In each case, models were generated with 1–30 factors. For each of these models, all combinations of mean centering and spectral normalization were tested which resulted in four models with each factor (no mean centering or normalization; mean centering without normalization; normalization without mean centering; mean centering with normalization). The spectral ranges tested for each analyte included the whole 4850–4250 cm^{-1} range and several more narrowly defined ranges which were selected on the basis of the positions of known absorption bands for a particular analyte (see Fig. 4). Additional ranges used for cholesterol, as an example, are 4500–4250 and 4450–4300 cm^{-1} which encompass the large absorbance features of cholesterol located in the low frequency region of the spectrum. Each spectral range tested is listed in Tables 4 and 5 which summarize conditions and results from the best PLS models with raw and Fourier filtered spectra, respec-

Solution – Print to Postscript

- If you're lucky, then the data is in vector notation within the pdf
- Example:

```
{
17 204 m
97 204 l
126 81 214 -12 361 -12 c
440 -12 517 17 578 62 c
554 109 l
501 70 434 43 366 43 c
266 43 184 101 154 204 c
380 204 l
```

- Note: You don't need to know postscript – just look inside until the numbers look right

Solution – Print to Postscript

Vectors cont'd

- Step 1: Use vi to edit out the stuff that interests you
- Step 2: Write a perl one liner to convert to a comma separated text file
- Step 3: Load into excel or equivalent
- Step 4: Your data is now in page coordinates – use "trend" or equiv to fit to original axis

Postscript tricks

- Print the page of interest to postscript
- Use a PS viewer (not gs). (gsview, gswin)
    - Put the PS viewer in autoupdate mode
- View your PS with gvim
- Delete sections, and view
- If the graph of interest disappears – voila
- Else, undo, save, repeat

Solution – Print to Postscript

- If you're less lucky, then the data is in vector notation within the pdf
- You can decode the ASCII-85 to the original (jpg or CCITT group4 encoding)
- Or you can use gs

```
gs -dNOPAUSE -dSAFER -r8000x8000
  -sDEVICE=png16 -sOutputFile=out
  inputfile </dev/null
```

- Image tag, width, height

- ASCII-85 encoding of image data

```
<<
/ImageType 1
/Width 2899
/Height 3774
/ImageMatrix [2899 0 0 -3774 0 3774]
/BitsPerComponent 1
/Decode [0 1 ]
/_Filters [currentfile /ASCII85Decode filter dup <</K -1
/Columns 2899 >> /CCITTFaxDecode filter dup
] >>
pdf_imagemask
s2sWNmfO:jdf^[ikVuQ:?Z8$b;$GiV&3MEJL'\4qKu8#3\odG)qk3uQJjSo69@rql
mb5.RijJg$df3%rqd4i+s8W,[Od?2+FC>tTObQBO#*7u&88.!MSd8j.Ul@t\-o.S$
A]#p:W)])fCBYDi.Jb0(#!D<qk8aCobAnI%l<b5cSBsV56Qf7IYX[[bE[25%,.SH0
```

Get points from image

- Write mini-perl web server: (shown on next slide)

- Problem: Image is too big
  - Won’t work with IE (not a big issue)
  - Need to scroll to image each time

```
use IO::Socket;
$image= "c:/home/adler/work/2002/biopeak/spectra-enlarged.gif";
$server = IO::Socket::INET->new( Proto     => 'tcp',
                                 LocalPort => $PORT,
                                 Listen    => 5,
                                 Reuse     => 1);
die "can't setup server" unless $server;
while ($client = $server->accept()) {
    while (<$client>) {
        s/[\012\015]+$//;
        print OF "$1,$2; #x,y\n"
           if /^GET .*posn.x=(\d+)&posn.y=(\d+).*/;
        last if /^$/;
    }
    print $client qq{HTTP/1.1 200 OK
Content-Type: text/html

<HTML><TITLE>GET POINTS</TITLE>
    <BODY><FORM method="GET" action="http://localhost:9001/">
    <input type="image" name="posn" src="file:///$image" />
    </H1></BODY></HTML>\n\n};
    close $client;
}
```

Get points from image

- Note that irfanview puts XY posn in title:

- Idea: write a busy-wait loop to check for changes in title and write that to a file.
- Click on image points
- Load text file into excel or equivalent

```
use Win32::GuiTest qw(FindWindowLike GetWindowText );

$Win32::GuiTest::debug = 0; # Set to "1" to enable
verbose mode
my @windows = FindWindowLike(0, "- IrfanView", "");
my $win= $windows[0];

for (@ARGV) { print "$_\n" }
print STDERR "Click on 1,1 to quit";
my ($yy,$xx)= (0,0);
while (1) {
    select undef,undef,undef, 0.01;
    my $text = GetWindowText($win);
    next unless $text =~ /^XY:\((\d+),(\d+)\)/;
    next if $xx==$1 and $yy==$2;
    $xx=$1; $yy=$2;

    exit 0 if $xx==1 and $yy==1;

    print "$xx , $yy\n";

}
```