

# **Heavy Species in ADAS**

## **from the viewpoint of one lowly ion**

Martin O'Mullane

Department of Physics  
University of Strathclyde

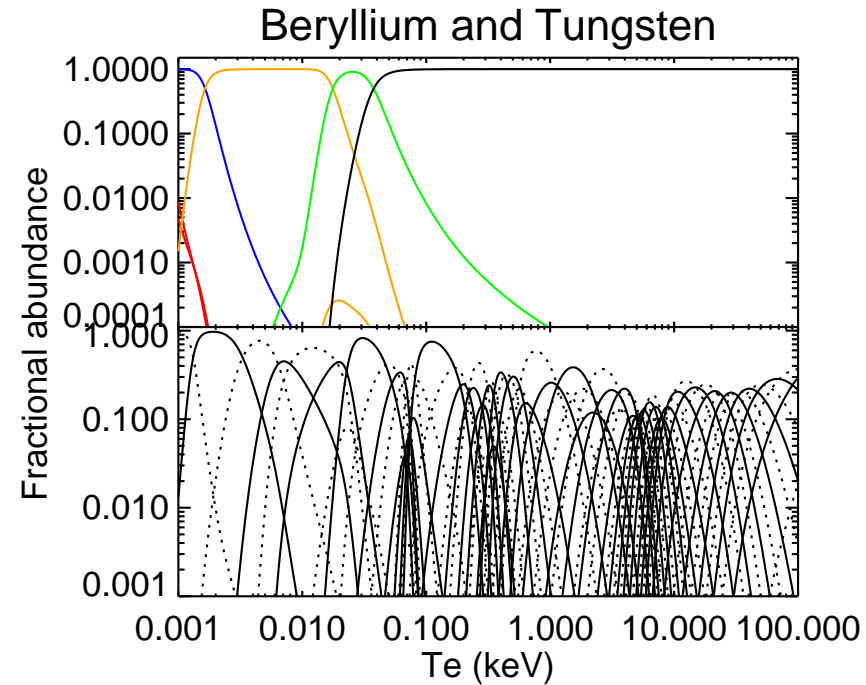
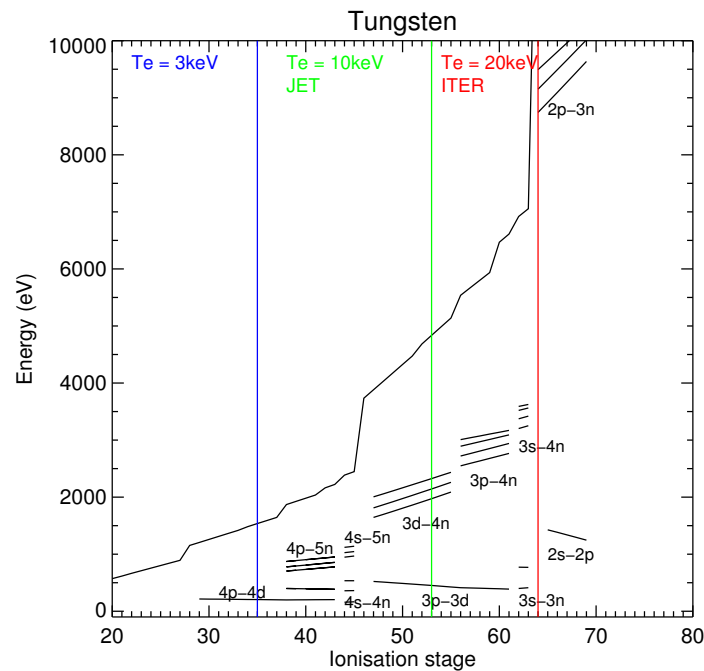
# Atomic data requirements

**FIZ CHEMIE BERLIN**  
Fachinformationszentrum Chemie GmbH

**The ultra-long Periodic Table of the Elements**

CERTIFIED ISO 9001 QUALITY SYSTEM

\* Stable isotope not known



# Three aspects of the heavy species question

## The data we need

- ▶ Source functions — *adf11* acd, scd, ccd
- ▶ Power coefficients — *adf11* plt, prb
- ▶ Line emission — *adf15* pec
- ▶ Spectral feature emission — *adf40* f-pec

## How to get it

- ▶ Scoping the problem
- ▶ Automated generation

## How to use it

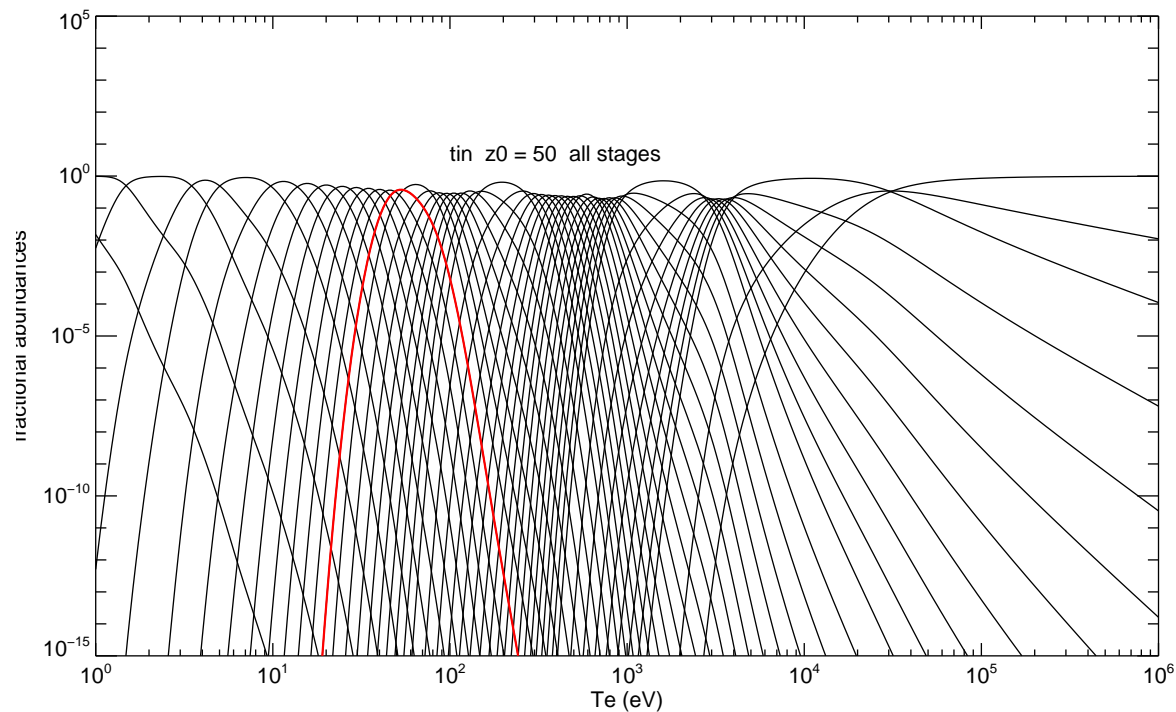
- ▶ Potentially large datasets
- ▶ Partitioning and superstaging.

## Let's choose tin (Z=50)

**First questions:** Where do its stages radiate? And what if there is no helpful ADAS *adf11* data?

```
preview_natural_partition, z0=50, plot_type=3, $  
te_min=1, te_max=1e6, $  
te_plot=1000, $  
frac=frac
```

```
oplot, frac.te, frac.ion_all[13,*], color=5, thick=5
```



## Now let's narrow our focus to Sn<sup>13+</sup>

- ▶ What is its ground state configuration?
- ▶ What configurations contribute to spectral emission?
- ▶ And to radiated power?
- ▶ How do we choose which ones to include?

The *adf00* set archives ionisation potential and ground configurations:

<code>tin</code>			<code>-50</code>										
<code>0</code>	<code>7.343d+00</code>	<code>1s2</code>	<code>2s2</code>	<code>2p6</code>	<code>3s2</code>	<code>3p6</code>	<code>3d10</code>	<code>4s2</code>	<code>4p6</code>	<code>4d10</code>	<code>4f0</code>	<code>5s2</code>	<code>5p2</code>
<code>1</code>	<code>1.463d+01</code>	<code>1s2</code>	<code>2s2</code>	<code>2p6</code>	<code>3s2</code>	<code>3p6</code>	<code>3d10</code>	<code>4s2</code>	<code>4p6</code>	<code>4d10</code>	<code>4f0</code>	<code>5s2</code>	<code>5p1</code>
				<code>..</code>									
				<code>..</code>									
<code>12</code>	<code>2.744d+02</code>	<code>1s2</code>	<code>2s2</code>	<code>2p6</code>	<code>3s2</code>	<code>3p6</code>	<code>3d10</code>	<code>4s2</code>	<code>4p6</code>	<code>4d2</code>			
<code>13</code>	<code>2.995d+02</code>	<code>1s2</code>	<code>2s2</code>	<code>2p6</code>	<code>3s2</code>	<code>3p6</code>	<code>3d10</code>	<code>4s2</code>	<code>4p6</code>	<code>4d1</code>			
<code>14</code>	<code>3.959d+02</code>	<code>1s2</code>	<code>2s2</code>	<code>2p6</code>	<code>3s2</code>	<code>3p6</code>	<code>3d10</code>	<code>4s2</code>	<code>4p6</code>				
					<code>..</code>								

# What configurations should be considered?

With a ground state of  $3d^{10}4s^24p^64d^1$  we can

- ▶ promote the valence 4d electron to any higher  $n/l$  shell
- ▶ allow 4s or 4p electrons to be excited
- ▶ or any other electron — from 2p perhaps?
- ▶ however where do we stop in  $\Delta n$  or  $\Delta l$ ?
- ▶ and how many configurations should we consider?

There are 180 distinct ground configurations (for elements up to Radon)

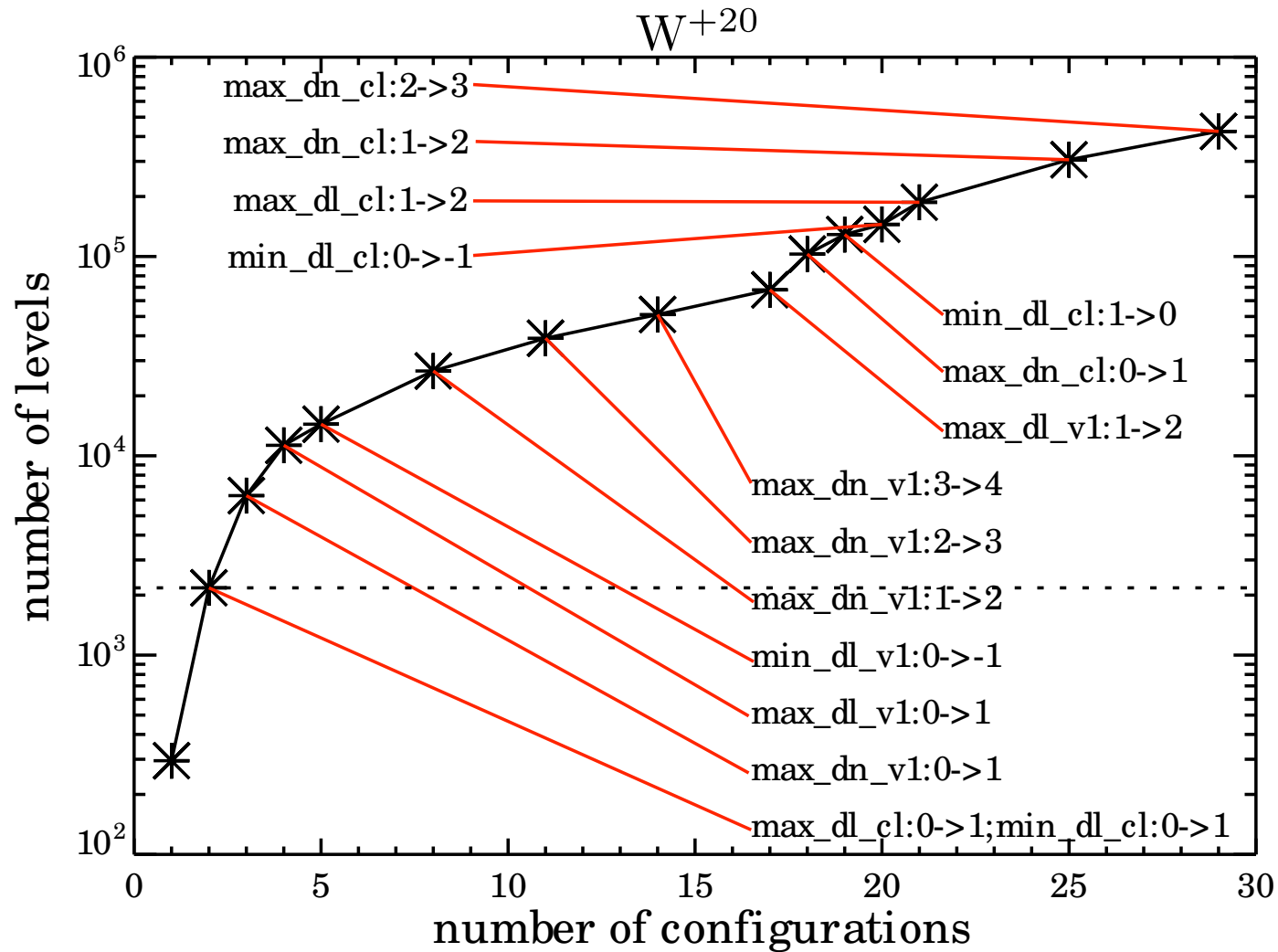
A rule based method is desirable (essential!)

# ADAS rules for choosing where to promote electrons

<i>index[]</i>	:	index of ground configuration of each ion of element in <i>adf54</i> file
<i>config[]</i>	:	ground configuration for each ion of element
<i>n_el[]</i>	:	number of electrons for each ion of element
<i>no_v_shl[]</i>	:	number of open (valence) shells. Include outer-most shell even if closed.
<i>max_dn_v1[]</i>	:	maximum $\Delta n$ promotion for first (outer-most) valence shell.
<i>min_dn_v1[]</i>	:	minimum $\Delta n$ promotion for first (outer-most) valence shell. Negative value allows access to inner unoccupied or open shells
<i>max_dl_v1[]</i>	:	maximum delta $\Delta l$ promotion for first (outer-most) valence shell.
<i>min_dl_v1[]</i>	:	minimum delta $\Delta l$ promotion for first (outer-most) valence shell.
<i>max_dn_v2[]</i>	:	maximum $\Delta n$ promotion for second (inner-most) valence shell.
<i>min_dn_v2[]</i>	:	maximum $\Delta n$ promotion for second (inner-most) valence shell.
<i>max_dl_v2[]</i>	:	maximum delta $\Delta l$ promotion for second (inner-most) valence shell.
<i>min_dl_v2[]</i>	:	minimum delta $\Delta l$ promotion for second (inner-most) valence shell.
<i>prom_cl[]</i>	:	promote from inner shell closed shells (1=yes,0=no).
<i>max_n_cl[]</i>	:	maximum inner shell <i>n</i> from which promotions are permitted.
<i>min_n_cl[]</i>	:	minimum inner shell <i>n</i> from which promotions are permitted.
<i>max_l_cl[]</i>	:	maximum inner shell <i>l</i> from which promotions are permitted.
<i>min_l_cl[]</i>	:	minimum inner shell <i>l</i> from which promotions are permitted.
<i>max_dn_cl[]</i>	:	maximum $\Delta n$ promotion from a permitted inner shell.
<i>min_dn_cl[]</i>	:	minimum $\Delta n$ promotion from a permitted inner shell. Negative values of $\Delta n$ allow access to inner unoccupied or open shells.
<i>max_dl_cl[]</i>	:	maximum $\Delta l$ promotion from a permitted inner shell.
<i>min_dl_cl[]</i>	:	minimum $\Delta l$ promotion from a permitted inner shell.
<i>fill_n_v1[]</i>	:	add all <i>nl</i> configurations of outer valence shell <i>n</i> (1=yes,0=no).
<i>fill_par[]</i>	:	if <i>n_fill</i> only add opposite parity to valence shell else add both parities (1=yes, 0=no).
<i>for_tr_sel[]</i>	:	Cowan option for radiative transitions 1 - first parity, 2 or 3(default).
<i>last_4f[]</i>	:	shift an electron valence shell to unfilled 4f as extra ground.
<i>grd_cmplx[]</i>	:	include configurations of same complex as ground configuration for valence <i>n</i> -shell.



# *adf54* : rules for automatic data generation



Care needed!! resolved calculations (ic or LS) can overwhelm computers.



## Work through Sn<sup>13+</sup>

- ▶ Within ADAS the generation of heavy species data is almost exclusively a non-GUI activity.
- ▶ The outputs are standard *adf11*, *adf15* and *adf40* datasets which can be used and examined with the GUI interactive system.

At the IDL command line:

```
; Let's choose Sn13+
```

```
z_nuc = 50
```

```
z_ion = 13
```

```
tag   = xxesym(z_nuc, /lower) + string(z_ion, format='(i2.2)')
```

```
; Use promotion rules from W work
```

```
a54file = '/home/adas/adas/adf54/promotion_rules_w_adf54.dat'
```

```
adas8xx_promotion_rules, z0_nuc = z_nuc, z_ion = z_ion, ionpot = ip, $
    prom_rules=rules, a54file = file
help, rules, /st
```

```
** Structure <9b54e9c>, 25 tags, length=60, data length=60, refs=1:
CONFIG          STRING ' 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d1'
INDEX           INT           129
NO_V_SHL        INT           1
MAX_DN_V1       INT           3
MIN_DN_V1       INT           0
MAX_DL_V1       INT           2
MIN_DL_V1       INT          -2
MAX_DN_V2       INT           0
MIN_DN_V2       INT           0
MAX_DL_V2       INT           0
MIN_DL_V2       INT           0
PROM_CL         INT           1
MAX_N_CL        INT           4
MIN_N_CL        INT           4
```

MAX_L_CL	INT	1
MIN_L_CL	INT	0
MAX_DN_CL	INT	1
MIN_DN_CL	INT	0
MAX_DL_CL	INT	2
MIN_DL_CL	INT	0
FILL_N_V1	INT	1
FILL_PAR	INT	0
FOR_TR_SEL	INT	3
LAST_4F	INT	0
GRD_CMPLX	INT	0

```
adas8xx_promotions, z0_nuc = z_nuc, z_ion = z_ion, ionpot = ip, $
                    prom_rules      = rules,                    $
                    promotion_results = results
```

```
help, results, /st
```

```
** Structure <9b530dc>, 11 tags, length=2496, data length=2496, refs=1:
GRD_CFG          STRING      '4d1  '
GRD_OCC          INT         Array[36]
EX_CFG           STRING      Array[25]
GRD_PAR          INT         0
EX_PAR           INT         Array[25]
GRD_ZC_COW       LONG        -14
EX_ZC_COW        LONG        Array[25]
OC_STORE         INT         Array[36, 26]
NO_CONFIGS       LONG        Array[7]
NO_TERMS         LONG        Array[7]
NO_LEVELS        LONG        Array[7]
```

```
print, results.grd_occ
```

2	2	6	2	6	10	2	6	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0				

```
print, results.oc_store[* ,1]
```

2	2	6	2	6	10	2	6	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0				

```
print, results.oc_store[* ,2]
```

2	2	6	2	6	10	2	6	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0				

; Write CA driver files for restricted plasma parameters

```
files = { adf34_file      : 'adf34_ca_' + tag + '.dat', $
          adf42_ca_file  : 'adf42_ca_' + tag + '.dat', $
          adf04_ca_file  : 'adf04_ca_' + tag + '.dat', $
          adf40_ca_file  : 'adf40_ca_' + tag + '.dat', $
          adf15_ca_file  : 'adf15_ca_' + tag + '.dat', $
          adf11_ca_file  : 'adf11_ca_' + tag + '.dat' }
```

```
plasma = {theta          : [ 1.0e3, 2.0e3, 5.0e3, 1.0e4, 1.5e4, $
                           2.0e4, 5.0e4, 1.0e5], $
          indx_theta     : indgen(8),
          rho            : [ 1.0e8, 1.0e10, 1.0e12, 1.0e14],
          indx_rho       : indgen(4),
          npix           : [ 128, 256],
          wvlmin         : [100.0, 1.0],
          wvlmax         : [150.0, 500.0],
          indx_wvl       : indgen(2),
          theta_noscale  : 0,
          rho_scale      : 0
```

```
adas8xx_create_drivers, z0_nuc=z_nuc, z_ion=z_ion, ionpot=ip, $
                        promotion_results=results,          $
                        plasma=plasma, files=files
```

### The driver file for ADAS801 (Cowan code):

```
2  -5    2   10  1.0    5.d-09    5.d-11-2  0130    1.0 0.65  0.0  0.5
50  -14   Sn ground z1=13 0    4d1
50  -14   Sn cfg 01    0    5s1
50  -14   Sn cfg 02    0    5d1
50  -14   Sn cfg 03    0    5g1
50  -14   Sn cfg 04    0    6s1
50  -14   Sn cfg 05    0    6d1
50  -14   Sn cfg 06    0    6g1
50  -14   Sn cfg 07    0    7s1
50  -14   Sn cfg 08    0    7d1
50  -14   Sn cfg 09    0    7g1
50  -32   Sn cfg 10    0    3d10 4s1  4p6  4d2
50  -32   Sn cfg 11    0    3d10 4s1  4p6  4d1  5s1
```



50	-32	Sn	cfg	12	0	3d10	4s1	4p6	4d1	5d1
50	-32	Sn	cfg	13	0	3d10	4s2	4p5	4d1	4f1
50	-32	Sn	cfg	14	0	3d10	4s2	4p5	4d1	5p1
50	-32	Sn	cfg	15	0	3d10	4s2	4p5	4d1	5f1
50	-14	Sn	cfg	16	1	4f1				
50	-14	Sn	cfg	17	1	5p1				
50	-14	Sn	cfg	18	1	5f1				
50	-14	Sn	cfg	19	1	6p1				
50	-14	Sn	cfg	20	1	6f1				
50	-14	Sn	cfg	21	1	7p1				
50	-14	Sn	cfg	22	1	7f1				
50	-32	Sn	cfg	23	1	3d10	4s1	4p6	4d1	5p1
50	-32	Sn	cfg	24	1	3d10	4s2	4p5	4d2	
50	-32	Sn	cfg	25	1	3d10	4s2	4p5	4d1	5d1

-1

## Back to the IDL command line:

; Run the CA structure code

```
adas8xx_create_ca_adf04, z_ion,          $  
                        z_nuc,          $  
                        results.oc_store, $  
                        ionpot          = ip,          $  
                        plasma          = plasma,      $  
                        adf04_t3_file   = files.adf04_ca_file
```

# adf04 file for Sn<sup>13+</sup>

Sn+13	50	14	2415630.6							
1	19	(0)0(	4.5)	0.0						
2	606527558529	(0)0(	134.5)	604454.8						
3	1B	(0)0(	0.5)	656865.3						
4	1A	(0)0(	6.5)	664371.3						
5	1C	(0)0(	2.5)	810958.4						
6	1D	(0)0(	4.5)	1048671.2						
7	606517568529	(0)0(	44.5)	1052972.9						
8	60652755851951A	(0)0(	419.0)	1259268.0						
9	1E	(0)0(	6.5)	1290521.2						
10	1G	(0)0(	0.5)	1291113.7						
11	1H	(0)0(	2.5)	1366810.5						
12	1F	(0)0(	8.5)	1400752.2						
13	60652755851951C	(0)0(	179.5)	1412966.0						
14	1I	(0)0(	4.5)	1485688.9						
15	1J	(0)0(	6.5)	1609362.8						
16	1M	(0)0(	0.5)	1611744.1						
17	60652755851951D	(0)0(	299.5)	1649394.9						
18	1N	(0)0(	2.5)	1654541.6						
19	1K	(0)0(	8.5)	1668939.1						
20	60651756851951B	(0)0(	19.5)	1715621.3						
21	1O	(0)0(	4.5)	1722798.2						
22	1P	(0)0(	6.5)	1795180.9						
23	1Q	(0)0(	8.5)	1831229.6						
24	60651756851951C	(0)0(	59.5)	1867168.5						
25	60652755851951E	(0)0(	419.0)	1891442.4						
26	60651756851951D	(0)0(	98.5)	2103915.7						
-1										
14.0	3	1.96+05	3.92+05	9.80+05	1.96+06	2.94+06	3.92+06	9.80+06	1.96+07	
4	1	1.92+11	7.92+00	8.17+00	8.96+00	1.02+01	1.12+01	1.20+01	1.54+01	1.87+01
3	1	7.69+06	1.99-01	2.02-01	2.09-01	2.17-01	2.23-01	2.27-01	2.39-01	2.47-01
5	1	1.07+11	3.89-01	4.07-01	4.70-01	5.77-01	6.73-01	7.59-01	1.13+00	1.50+00
6	1	1.17+07	7.21-01	7.29-01	7.54-01	7.91-01	8.20-01	8.42-01	9.17-01	9.69-01
9	1	2.22+10	1.34-01	1.35-01	1.35-01	1.37-01	1.39-01	1.41-01	1.59-01	1.83-01
12	1	1.21+08	5.57-01	5.62-01	5.81-01	6.11-01	6.36-01	6.57-01	7.33-01	7.89-01
10	1	2.97+06	2.19-02	2.19-02	2.22-02	2.25-02	2.27-02	2.29-02	2.35-02	2.39-02

# Eissner notation — quick recap

Each occupation/orbital-list pair is separated from the next by 5 (or 6)

1s

1

2s 2p

2 3

3s 3p 3d

4 5 6

4s 4p 4d 4f

7 8 9 A

5s 5p 5d 5f 5g

B C D E F

etc.

# Generating spectral and power data — ADAS810

Process the *adf42* file made by `adas8xx_create_drivers` with ADAS810 to generate *adf11/plt*, *adf15* and *adf40* datasets.

The screenshot shows a dialog box titled "Driver File Details:-". At the top, there are two tabs: "Standard file (adf04)" and "Driver File (adf42)". The "Driver File (adf42)" tab is selected. Below the tabs, there is a "Data Root" text field. Underneath, there are two buttons: "Central Data" and "User Data", and a checkbox labeled "Edit Path Name". A large text area contains a list of files, with "adf42\_ca\_sn13.dat" selected. The list includes: ". .", "adf04\_ca\_sn13.dat", "adf11\_ca\_sn13.dat", "adf15\_ca\_sn13.dat", "adf34\_ca\_sn13.dat", "adf40\_ca\_sn13.dat", and "adf42\_ca\_sn13.dat". At the bottom of the dialog, there is a label "Enter driver file name" and three buttons: "Browse Comments", "Cancel", and "Done".

Title for Run [ ]

Nuclear Charge: 50 Ion Charge: 13

Input from adf42 file : ./adf42\_ca\_sn13.dat [Browse Comments](#)

adf04 file : adf04\_ca\_sn13.dat [Browse Comments](#)

No expansion data

### Temperatures

INDEX	Electron	Ion	Neutral Hydrogen	Input Value
1	1.689E+01	0.000E+00	0.000E+00	1.689E+01
2	3.378E+01	0.000E+00	0.000E+00	3.378E+01
3	8.444E+01	0.000E+00	0.000E+00	8.445E+01
4	1.689E+02	0.000E+00	0.000E+00	1.689E+02
5	2.533E+02	0.000E+00	0.000E+00	2.533E+02

Temperature Units: eV

[Edit Table](#)

Default: [Standard Set](#)

### Densities

INDEX	Electron Densities	Ion Densities
1	1.000E+08	0.000E+00
2	1.000E+10	0.000E+00
3	1.000E+12	0.000E+00
4	1.000E+14	0.000E+00
5		

Density Units: cm-3

[Edit Table](#)

Default: [Standard Set](#)

### Metastable State

- 19
- 606527558529
- 1B
- 1A

For a single metastable normalise PLT and PEC ?

- NO
- YES

### Include Reactions:

- Ionisation Rates
- Charge Exchange
- Recombination
- Inner Shell Ionisation
- Include Projection Data
- Proton Impact Collisions

Zeff :

### Spectral Intervals

INDEX	# pixels	min wave	max wave
1	128	100.00	150.00
2	256	1.00	500.00
3			
4			
5			

[Edit Table](#)

Lower limit of A-value:

Edit the processing options data and press Done to proceed

[Cancel](#) [Done](#)

Text Output    Replace   Default File Name

File Name :

---

PEC (adf15) file    Replace   Default File Name

File Name :

Feature PEC (adf40) file    Replace   Default File Name

File Name :


Total power (adf11/plt) unfiltered    Replace   Default File Name

File Name :

Total power (adf11/plt) filtered    Replace   Default File Name

File Name :

Choose output options





# Where lies the emission?

Back to the IDL command line!

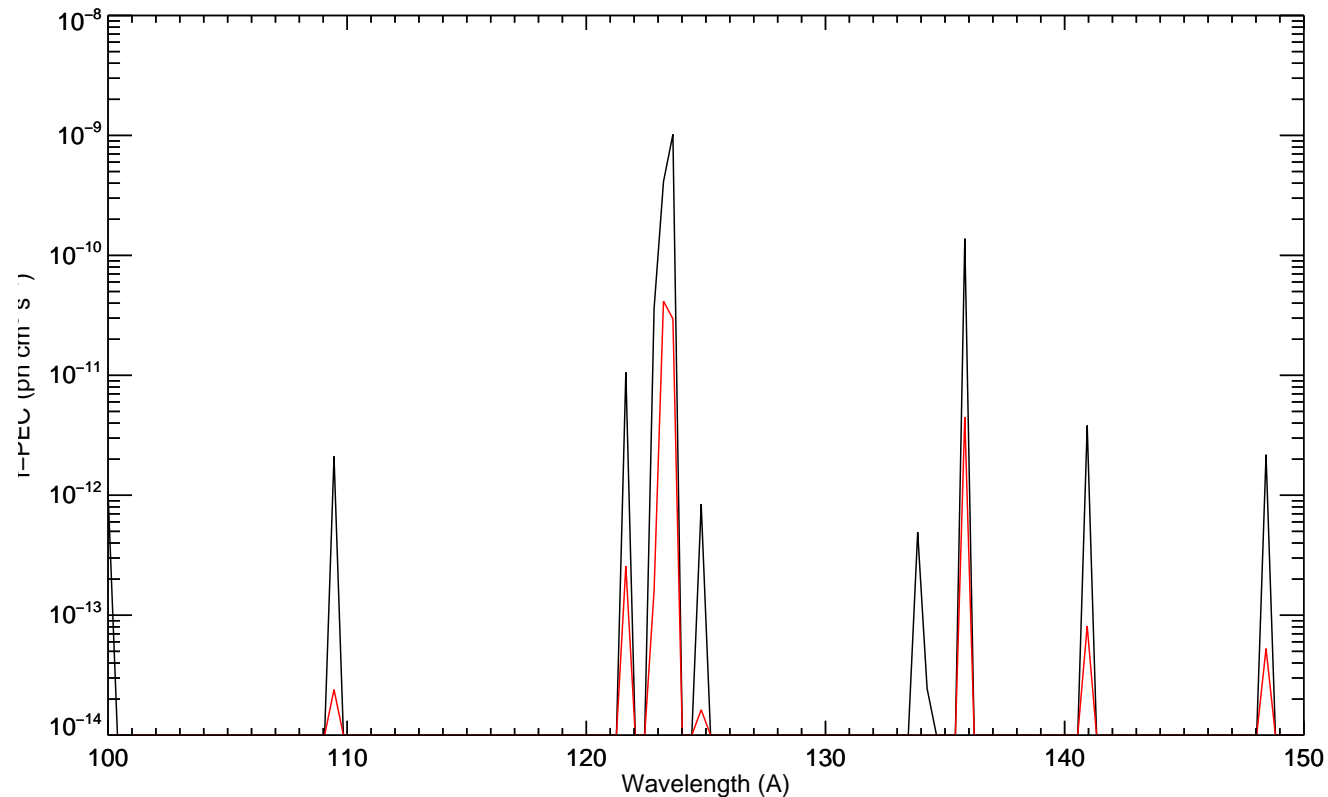
```
read_adf40,file='adf40_ca_sn13.dat', fulldata=all
```

```
help, all, /st
```

ESYM	STRING	'Sn'	
IZO	LONG		50
IS	LONG		13
IS1	LONG		14
NBLOCK	LONG		2
NPIX	LONG	Array [2]	
WAVE_MIN	DOUBLE	Array [2]	
WAVE_MAX	DOUBLE	Array [2]	
NTE	LONG	Array [2]	
TE	DOUBLE	Array [8, 2]	
NDENS	LONG	Array [2]	
DENS	DOUBLE	Array [4, 2]	
FPEC	DOUBLE	Array [256, 8, 4, 2]	
TYPE	STRING	Array [2]	

```
wave=adas_vector(low=all.wave_min[0], high=all.wave_max[0], $
                 num=all.npix[0], /linear)
```

```
plot_io, wave, all.fpec[* , 7, 2, 0] > 1e-14, $
      xtitle='Wavelength (A)', ytitle = 'f-PEC (ph cm!u3!n s!u-1!n)',
oplot, wave, all.fpec[* , 1, 2, 0] > 1e-14, color=5
```



# How to identify contributing configurations

It depends of the width of the spectral region of interest

C	lv	configuration	(2S+1)L(w-1/2)	energy (cm <sup>-1</sup> )					
C	1	19	(0)0( 4.5)	0.0					
C	2	606527558529	(0)0(134.5)	604454.8					
C	3	1B	(0)0( 0.5)	656865.3					
C	4	1A	(0)0( 6.5)	664371.3					
C	5	1C	(0)0( 2.5)	810958.4					
C	6	1D	(0)0( 4.5)	1048671.2					
C	7	606517568529	(0)0( 44.5)	1052972.9					
-									
C	20	60651756851951B	(0)0( 19.5)	1715621.3					
C	21	10	(0)0( 4.5)	1722798.2					
C	22	1P	(0)0( 6.5)	1795180.9					
-									
C	12	94.9692	7(0)0( 44.5) - 1(0)0( 4.5)		excit	1	1	13	17 48 12
C	13	99.5453	19(0)0( 8.5) - 4(0)0( 6.5)		excit	1	1	13	33 14 13
C	14	100.233	18(0)0( 2.5) - 3(0)0( 0.5)		excit	1	1	13	44 43 14
C	15	109.668	21(0)0( 4.5) - 5(0)0( 2.5)		excit	1	1	13	54 39 15
C	16	121.756	14(0)0( 4.5) - 4(0)0( 6.5)		excit	1	1	13	31 28 16
C	17	122.821	24(0)0( 59.5) - 7(0)0( 44.5)		excit	1	1	13	117 18 17
C	18	123.311	5(0)0( 2.5) - 1(0)0( 4.5)		excit	1	1	13	3 6 18
C	19	123.684	13(0)0(179.5) - 2(0)0(134.5)		excit	1	1	13	124 5 19
C	20	124.877	16(0)0( 0.5) - 5(0)0( 2.5)		excit	1	1	13	52 44 20

- ▶ Overplot/look at PEC *adf15* data.
- ▶ Refine promotion rules or *adf34* driver to home-in
- ▶ Note that structure codes are not spectroscopically accurate.

## Where next?

Identify emission region of interest — treat these in intermediate coupling.

For Sn<sup>13+</sup> :

- ▶ 26 configurations, 226 terms, 554 levels

```
read_adf40,file='fpec40#sn_ca#sn13.dat', fulldata=all_ca
```

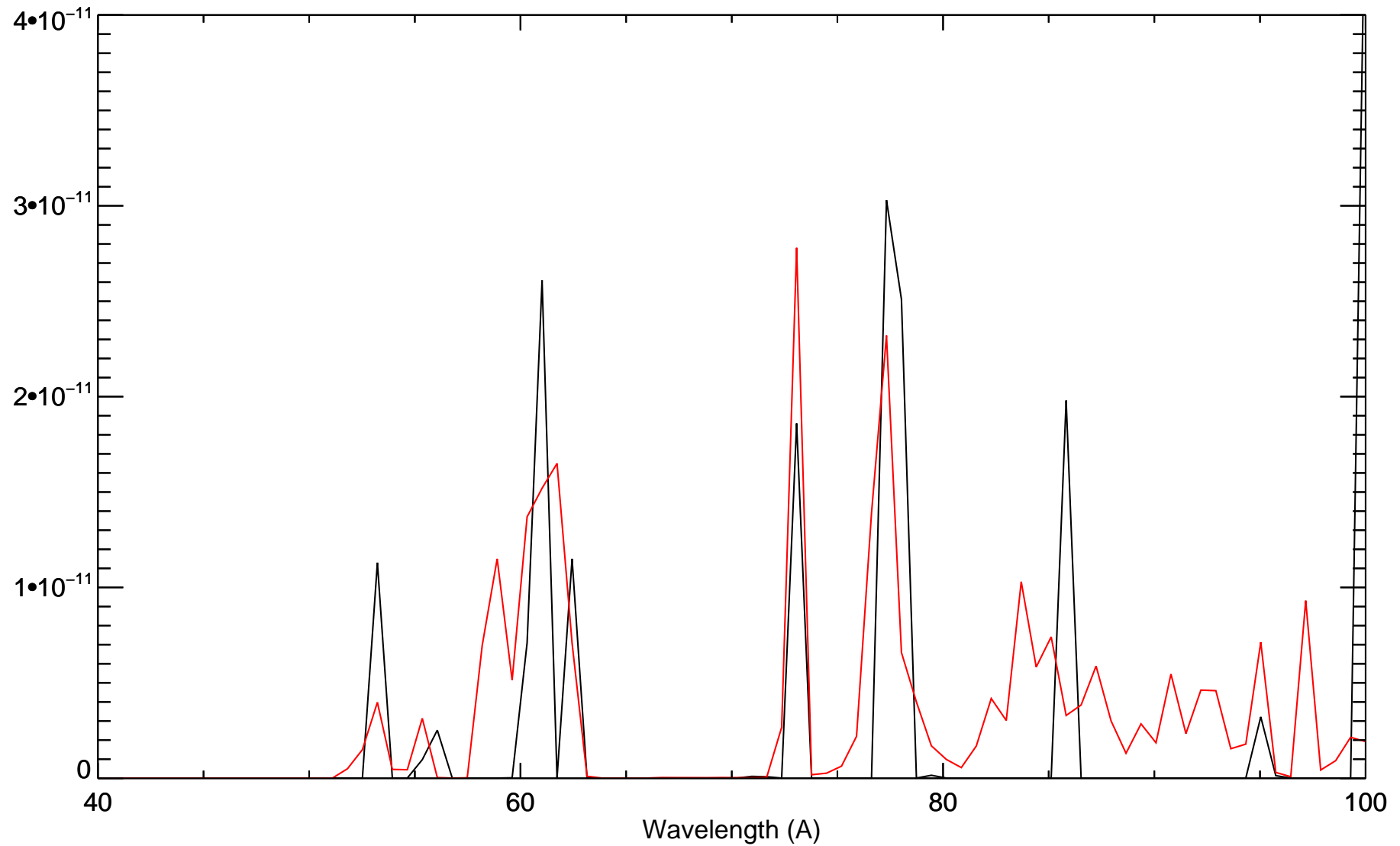
```
read_adf40,file='fpec40#sn_ic#sn13.dat', fulldata=all_ic
```

```
wave=adas_vector(low=all_ca.wave_min[1], high=all_ca.wave_max[1], $  
                 num=all_ca.npix[1], /linear)
```

```
plot, wave, all_ca.fpec[* , 7, 2, 1] > 1e-14,      $  
      xtitle = 'Wavelength (A)',                  $  
      ytitle = 'f-PEC (ph cm3 n s-1 n)',          $  
      xrange = [40, 100], yrange = [1e-12, 4e-11]
```

```
oplot, wave, all_ic.fpec[* , 7, 2, 1] > 1e-14, color=5
```

## Configuration average (CA) vs intermediate coupling (ic)



# In Reality

- ▶ Do not consider each stage by hand — by element is preferable.
- ▶ Many scripts available within ADAS to aid this task.
- ▶ These impose a directory structure.
- ▶ Baseline data identified by the year '40' tag.
- ▶ Full instructions in forthcoming ADAS technical report.

# Ionisation

Very similar to excitation — driven by *adf56* collection of rules

<i>index[]</i>	:	index of ground configuration of each ion of element in <i>adf56</i> file
<i>config[]</i>	:	ground configuration for each ion of element
<i>n_el[]</i>	:	number of electrons for each ion of element
<i>no_v_shl[]</i>	:	number of shells to treat as valence shells. Max. 2 relevant to relating ion and parent.
<i>v1_shl[]</i>	:	first valence shell position in <i>adf56</i> configuration specifications.
<i>v2_shl[]</i>	:	second valence shell position in <i>adf56</i> configuration specifications. zero if none defined.
<i>drct_eval_v[]</i>	:	evaluate direct ionisation from the valence shell(s).
<i>drct_eval_cl[]</i>	:	evaluate direct ionisation from other non-valence (closed) shells.
<i>min_shl_cl[]</i>	:	lowest closed shell to include (position in <i>adf56</i> configuration specifications).
<i>exca_eval_v2[]</i>	:	evaluate excitation/autoionisation from second valence shell if identified.
<i>max_dn_v2[]</i>	:	maximum change in v2 n-shell to be included.
<i>min_dn_v2[]</i>	:	minimum change in v2 n-shell to be include.
<i>max_dl_v2[]</i>	:	maximum change in v2 l-shell to be included.
<i>min_dl_v2[]</i>	:	minimum change in v2 l-shell to be include.
<i>exca_eval_cl[]</i>	:	evaluate excitation/autoionisation from other non-valence (closed) shells.
<i>max_dn_cl[]</i>	:	maximum change in closed n-shell to be included.
<i>min_dn_cl[]</i>	:	minimum change in closed n-shell to be included.
<i>max_dl_cl[]</i>	:	maximum change in closed l-shell to be included.
<i>min_dl_cl[]</i>	:	minimum change in closed l-shell to be included.
<i>exst_eval[]</i>	:	evaluate ionisation from excited states.
<i>exst_adf00_prt[]</i>	:	assume parent for building excited states is as present in the <i>adf00</i> data set for the ion.
<i>exst_prt_hole_shl[]</i>	:	specify position of shell in ground configuration to form parent if not from <i>adf00</i> above.
<i>max_n_exst[]</i>	:	maximum n-shell for excited states to be included.
<i>max_l_exst[]</i>	:	maximum l-shell for excited states to be included.
<i>drct_eval_exst_v[]</i>	:	evaluate direct ionisation from excited state valence shells.
<i>drct_eval_exst_cl[]</i>	:	evaluate direct ionisation from excited state non-valence (closed) shells.
<i>exca_eval_exst_v[]</i>	:	evaluate excitation/autoionisation for excited states from valence shells (v1 and v2 above).
<i>exca_eval_exst_cl[]</i>	:	evaluate excitation/autoionisation for excited states from non-valence (closed) shells.

*adf32* is the driver file for CADW ionisation code from the Auburn group.



## Once again to the IDL command line!

```
; Add offline-ADAS IDL library to the path

!path = expand_path('/home/adas/offline_adas/adas8#2/idl') + ':' + !path

; Promotion rules - compiled by Adam Foster (arf)

a56file = '/home/adas/adas/adf56/large_arf09.dat'

; Sn13+ !!

adas8xx_ionis_promotion_rules, z_nuc      = 50,          $
                                z_ion      = 13,          $
                                a56file     = a56file,     $
                                adf32      = 'adf32_ca_sn13.dat', $
                                comments    = ['C-----', $
                                              'C  I made this!', $
                                              'C-----']
```

```

elem   = Sn
stage  = 13
ip_z   = 3193147.3
ip_z1  = 2415629.2
seq    = rb
-----
Type = Direct /number=3/
#
200-51 1 2 01. 1. 5.0E-08 1.0E-11-2 0130 0 1.00 0.65 71. 0.5 0.70
50 14 sn+13 ground 4d1 4d
50 15 sn+14 from 4d 3d10 4s2 4p6
-1

#
200-51 1 2 01. 1. 5.0E-08 1.0E-11-2 0130 0 1.00 0.65 73. 0.5 0.70
50 14 sn+13 ground 4d1 4s
50 15 sn+14 from 4s 3d10 4s1 4p6 4d1
-1

#
200-51 1 2 01. 1. 5.0E-08 1.0E-11-2 0130 0 1.00 0.65 72. 0.5 0.70
50 14 sn+13 ground 4d1 4p
50 15 sn+14 from 4p 3d10 4s2 4p5 4d1
-1
-----
Type = InDirect /number=2/
#
20 -51 0 2 10 1.0 5.e-08 1.e-11-2 130 1.0 0.65 66. 0.5 0.7
50 14 sn+13 ground 4d1 4s
50 14 sn+13 via 4d 3d10 4s1 4p6 4d2 4d
50 14 sn+13 via 4f 3d10 4s1 4p6 4d1 4f1 4f
50 14 sn+13 via 5s 3d10 4s1 4p6 4d1 5s1 5s
-
-
50 14 sn+13 via 7h 3d10 4s1 4p6 4d1 7h1 7h
50 14 sn+13 via 7i 3d10 4s1 4p6 4d1 7i1 7i
-1

#
20 -51 0 2 10 1.0 5.e-08 1.e-11-2 130 1.0 0.65 66. 0.5 0.7
50 14 sn+13 ground 4d1 4p
50 14 sn+13 via 4d 3d10 4s2 4p5 4d2 4d
50 14 sn+13 via 4f 3d10 4s2 4p5 4d1 4f1 4f
-
-
50 14 sn+13 via 7h 3d10 4s2 4p5 4d1 7h1 7h
50 14 sn+13 via 7i 3d10 4s2 4p5 4d1 7i1 7i
-1
-----
C-----
C I made this!
C-----

```

## Next to the unix command line

```
/home/adas/offline_adas/adas8#2/adas8#2.pl adf32_ca_sn13.dat sn13.dat
```

## Return to IDL to inspect the results

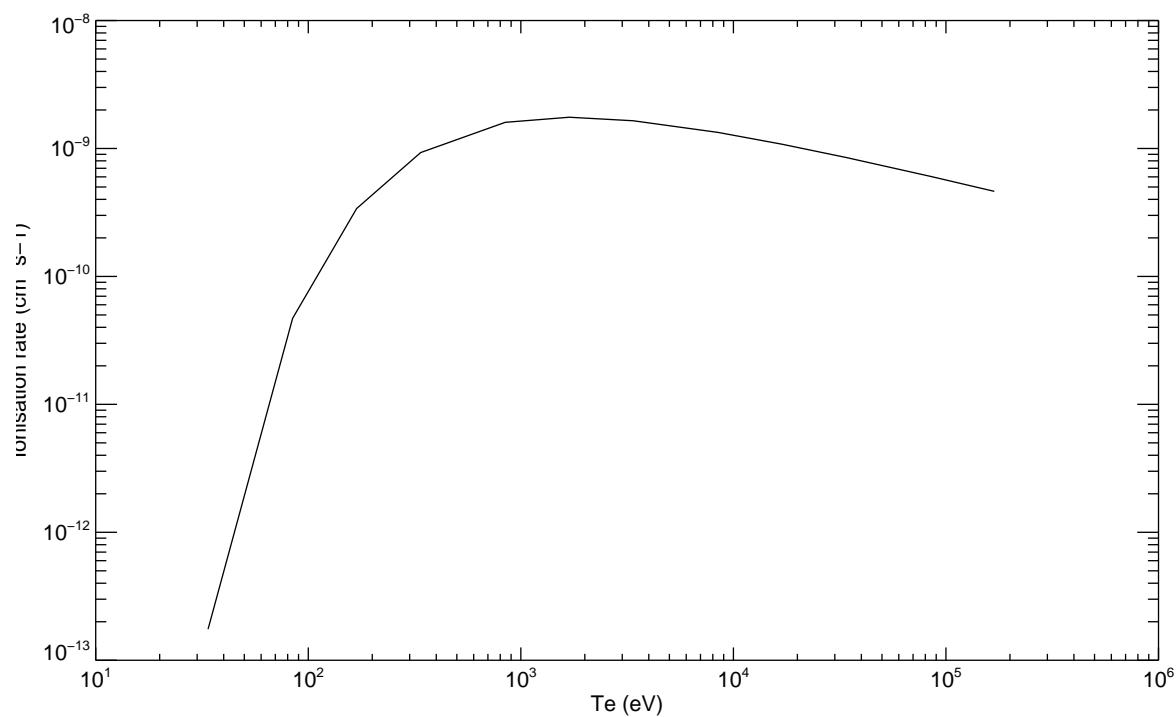
```
read_adf23, file='sn13.dat', fulldata=all, szd_total=szd
```

```
help, szd,/st
```

```
** Structure <a3e784c>, 7 tags, length=6576, data length=6576, refs=1:  
  TE          DOUBLE      Array[12]  
  Q_ION       DOUBLE      Array[1, 3, 12]  
  IS_Q_ION    LONG        Array[1, 3, 12]  
  Q_EXC       DOUBLE      Array[1, 41, 12]  
  IS_Q_EXC    LONG        Array[1, 41, 12]  
  QTOT        DOUBLE      Array[1, 1, 12]  
  IS_QTOT     LONG        Array[1, 1, 12]
```

```
te = reform(szd.te) / 11605.0
szd = reform(szd.qtot*10.0^szd.is_qtot) > 1.0e-36
```

```
plot_oo, te, szd, $
      xtitle='Te (eV)', $
      ytitle = 'Ionisation rate (cm3 s-1)'
```



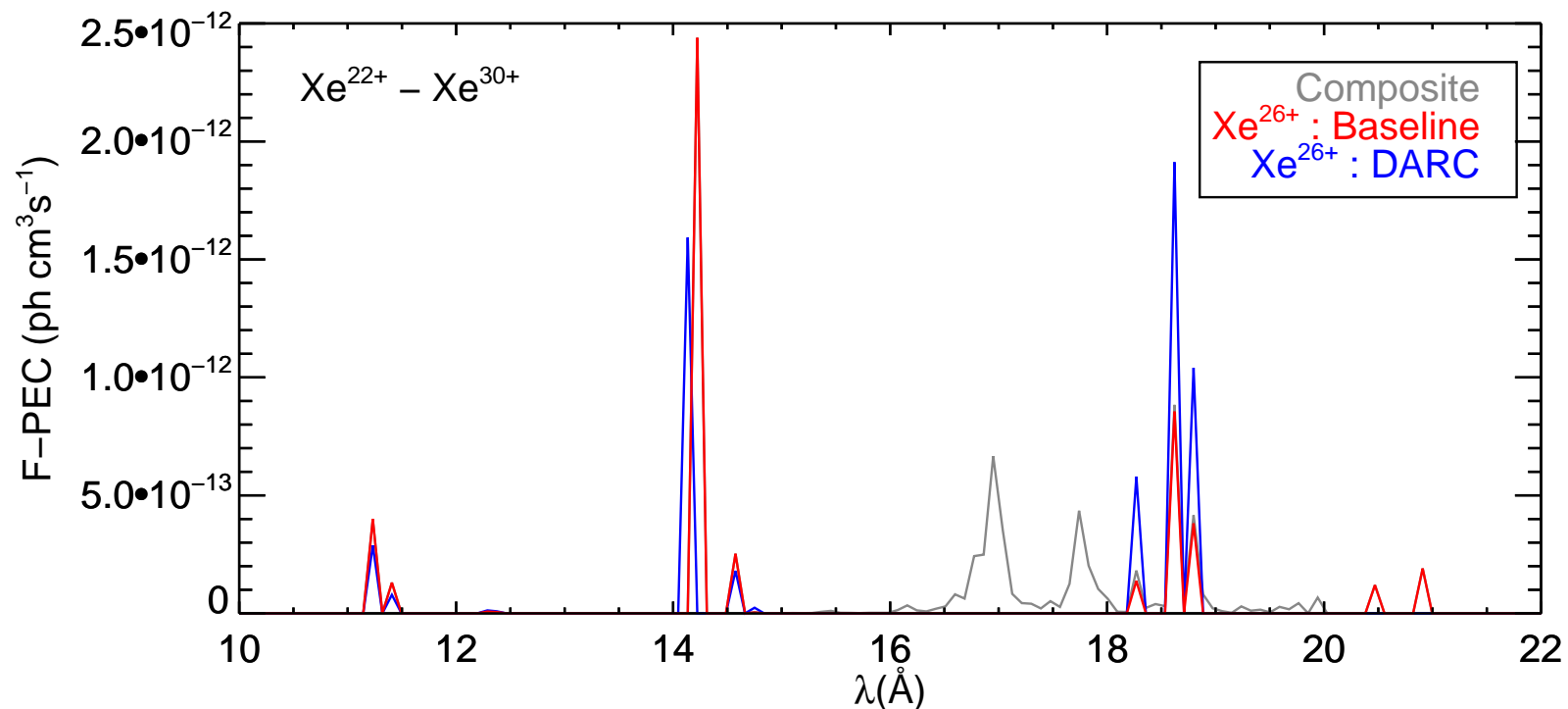
# Recombination

- ▶ Extension of BBGP program from DR is under active development.
- ▶ A set of *adf55* rules will follow.
- ▶ However, use ADAS407/ADAS408 for now.

# Selectively uplift the quality of baseline

- ▶ With increasing atomic number relativistic effects assume a greater importance.
- ▶ Compare the baseline Born data to DARC to assess its validity.

Consider Ni-like  $\text{Xe}^{+26}$  with a  $3d^{10}$  ground configuration:



## Handling heavy species data

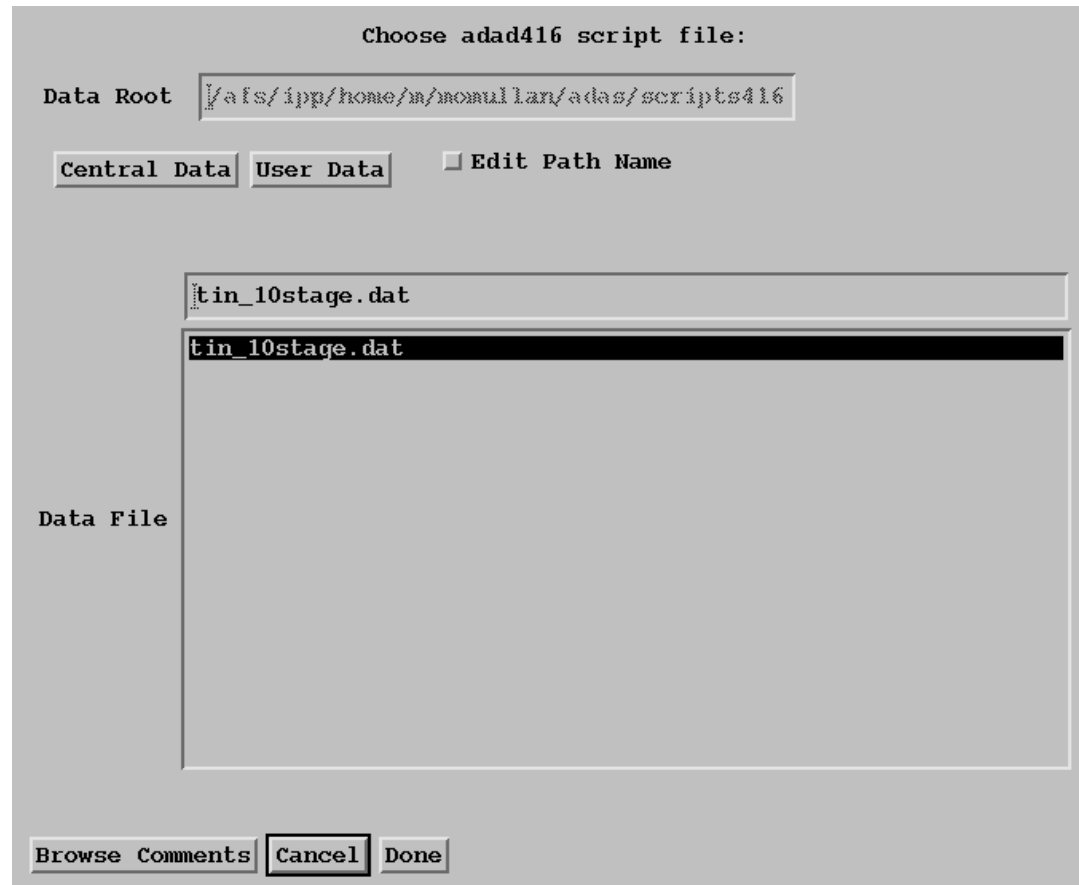
It may not always be necessary to consider all ionisation stages of an element.

Again, for tin, consider the partition (extract from *scripts416* driver file:

```
//#02/p00/ 00/  
  p01/ 01 02 03 04 05 06 07/  
  p02/ 08/  
  p03/ 09/  
  p04/ 10/  
  p05/ 11/  
  p06/ 12/  
  p07/ 13/  
  p08/ 14/  
  p09/ 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32  
 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49/  
  p10/ 50/
```

# Generate partitioned *adf11* data

Process with ADAS416 : See /home/g08190/scripts416/tin\_10stage.dat



Copy the *adf11* datasets from /home/g08190/adas/adf11/.

**Note: They will disappear by Friday!**



# Compare ionisation equilibrium balance

## At the IDL command line

```
te = adas_vector(low=1, high=1000, num=40)
dens = fltarr(40) + 1e12
```

```
; Explicitly name adf11 files
```

```
files = {acd : 'acd66_sn#10stage.dat', $
         scd : 'scd66_sn#10stage.dat' }
```

```
run_adas405, uid='adas', elem='sn', year=89, te=te, dens=dens, $
           files=files, frac=frac_par
```

```
files = {acd : 'acd89_sn.dat', $
         scd : 'scd89_sn.dat' }
```

```
run_adas405, uid='adas', elem='sn', year=89, te=te, dens=dens, $
           files=files, frac=frac
```

```
xmin = min(te, max=xmax)
ymin = 0.001
ymax = 1.5

plot_oo, [xmin, xmax], [ymin, ymax], /nodata, ystyle=1, $
    xtitle = 'Te (eV)', ytitle = 'Fractional abundance'

for j = 0, n_elements(frac.stage)-1 do begin
    oplot, te, frac.ion[* ,j]
endfor

for j = 0, n_elements(frac_par.stage)-1 do begin
    oplot, te, frac_par.ion[* ,j], color=5, thick=5
endfor
```

We assume that we have no great interest outside our chosen ions!

