**Documento explicativo para el procesado de los Langleys. Brewer / Dobson métodos**

Se explica a continuación cual es el proceso y cuales las funciones involucradas en el análisis de los Langleys para el Brewer. Se trabaja con dos metodologías ligeramente diferentes, correspondientes a los métodos Brewer y Dobson.

* Brewer Algorithm :

Siendo , , βi Rayleigh coefficient, m3 ozone airmass y m2 Rayleigh airmass. Si asumimos Ozono constante, entonces podremos hacer ,  es decir, la ETC será el corte con el eje de ordenadas para m3=0.

* Dobson algorithm:

En este caso se aplica la misma 1/m3 estrategia usada por los Dobsons, resultando

Lo que se obtiene ahora, a través de la pendiente ETCc, es la corrección a la ETC **estimada** ETCo. Las ventajas del método Dobson frente al método Brewer serían:

1. Las observaciones están mejor distribuidas frente a 1/m3 que frente a m3.
2. El error cometido al determinar la pendiente es inferior al error cometido al determinar el corte en m3=0.

En principio trabajaremos con las medidas sincronizadas entre los instrumentos. El tiempo de sincronización es Tsync = 5 minutos. Las medidas usadas son entonces los sumarios. Igualmente se podría trabajar con las medidas individuales (5 por cada sumario). Esto es así porque se ha respetado igual estructura para las matrices de datos independientemente de que datos son los usados, como se muestra a continuación. En el caso de las medidas individuales no se trata de datos simultáneos.

Asumiendo que vamos a trabajar con datos simultáneos, el punto de partida serán los sumarios recalculados por la función test\_recalculation.m. Inicialmente se optó por esta solución porque obtenía también el ozono, que luego se usaba para descartar días en los que el ozono no fuera constante. También se podrá trabajar con la variable ***ozone\_ds***, pues al cargar los datos (función readb\_ds\_develop.m -> read\_bdata.m) tenemos, entre otras, las siguientes salidas:

|  |  |
| --- | --- |
| **ozone.dsum** | Salidas raw, tomadas directamente del fichero B (sumarios) -> B config (1). Ratios. |
| **ozone.raw0** | Salidas raw, tomadas directamente del fichero B (med. individuales) -> B config (1). Ratios & counts recalc. |
| **ozone.ozone\_sum** | Summaries, recalculated (2 configs). No MS9`s, No raw counts |
| **ozone.ozone\_ds** | Ind. measurements, recalculated (2 configs). Ratios & MS9’s |
| **ozone.raw** | Ind. measurements. Raw & Recalc. counts, no Rayleigh (2 configs) |

En el caso de trabajar con ***ozone\_ds*** ya no tendremos el ozono para las dos configuraciones, sólo las MS9’s. Vemos en todo caso que no es directo obtener una variable con los sumarios que contenga las ratios MS9 y además las cuentas recalculadas (o brutas) para cada canal. La alternativa será trabajar con la función ya conocida langley\_data\_cell.m, calcular de aquí los sumarios (ya implementado), y a partir de aquí obtener las medidas simultáneas (ahora si dispondríamos de toda la información que se quiera: cuentas brutas, cuentas recalculadas, ratios, ozono, …). Todo esto quedará más claro a medida que se lea el documento que nos ocupa. En la figura abajo se muestran como ejemplo los resultados de los Langleys obtenidos de los sumarios calculados a partir de la función test\_recalculation.m (círculos) y los obtenidos de los sumarios recalculados a partir de las medidas individuales (función langley\_data\_cell.m, estrellas). Como se muestra, los resultados en ambos casos son idénticos (las dos únicas diferencias observadas son debidas a que el número de datos simultáneos es diferente, menor en todo caso).

General_.tif

Figure 1: Resultado del análisis de los Langley Plots sobre dos conjuntos de datos diferentes: a) MS9 simultáneas para la triada, obtenidas a partir de los sumarios recalculados por la función test\_recalculation.m (círculos) y b) MS9 obtenidas a partir de los sumarios recalculados directamente a partir de las variables ozone\_raw y ozone\_ds. En este caso no son medidas simultáneas. Método Brewer (arriba) y Dobson (abajo).

No voy a explicar aquí como se obtienen las variables summary y summary\_old, pero si conviene tener en cuenta lo siguiente:

1. **Aplicamos la función filter\_corr.m a los sumarios**. Entonces tendremos MS9 con y sin corrección por filtros, campos 8 y 9 de summary, respectivamente.
2. **No aplicamos la función filter\_corr.m a los sumarios**. Entonces tendremos MS9 sin corrección por filtros (campos 8). El campo 9 es ahora la desviación estándar

Por defecto se trabaja con el campo 8, de modo que la forma de trabajo preferida será por defecto aplicar siempre la función filter\_corr.m, y entonces trabajar bien con ***summary***/***summary\_old*** (filter corrected) bien con ***summary\_orig***/***summary\_orig\_old*** (no filter corrected).

Bien, una vez tenemos las variables ***summary***/***summary\_old***, se las pasaremos como input a la función langley\_summ\_sync.m. En resumen, esta función recibe como input las variables ***summary*** y ***summary\_old***, entre otros, y devuelve una celda para cada instrumento (**en Cal.n\_ref, corregir?**). Cada una de estas celdas contiene a su vez una celda para cada día conteniendo una matriz de datos de 39 columnas con la siguiente estructura:

Tabla 1: Salida de la función langley\_summ\_sync.m

|  |
| --- |
| ***time, lat, long, sza, m3, m2, flag,  NaN, tst, filt, temp*** ... % 1-11  ***NaN, NaN, NaN, NaN, NaN, NaN, NaN***  ... % 12-18 count-rates recalculated 1 (Rayleigh uncorrected !!)  ***O3 old, NaN, NaN, NaN, NaN, NaN, MS9 old***  … % 19-25 ratios recalculated 1 (Rayleigh corrected !!)  ***NaN, NaN, NaN, NaN, NaN, NaN, NaN*** ... % 26-32 count-rates recalculated 2 (Rayleigh uncorrected !!)  ***O3 old, NaN, NaN, NaN, NaN, NaN, MS9 old***  ... % 33-39 ratios recalculated 2 (Rayleigh corrected !!) |

Se ha mantenido esta estructura porque es la estructura de datos con la que se trabaja en el caso de la función langley\_data\_cell.m.

Tabla 2: Salida de la función langley\_data\_cell.m

|  |
| --- |
| ***'date' 'hg\_id' 'nds' 'sza' 'm2' 'm3' 'sza' 'saz' 'tst' 'filt' 'temp'*** *...% 1-11*  ***'f0' 'f1' 'f2' 'f3' 'f4' 'f5' 'f6' ...*** *% 12-18 count-rates recalculated 1 (Rayleigh uncorrected !!)*  ***'o3\_1' 'r1' 'r2' 'r3' 'r4' 'r5' 'r6' ...*** *% 19-25 ratios recalculated 1 (Rayleigh corrected !!)*  ***'F0' 'F1' 'F2' 'F3' 'F4' 'F5' 'F6' ...*** *% 26-32 count-rates recalculated 2 (Rayleigh uncorrected !!)*  ***'O3\_2' 'R1' 'R2' 'R3' 'R4' 'R5' 'R6' ...*** *% 33-39 ratios recalculated 2 (Rayleigh corrected !!)* |

Lo que quiero resaltar aquí es que la estructura de los datos simultáneos (sumarios) y de los individuales es idéntica. Esto nos permitirá desarrollar las funciones de análisis de forma que sean comunes para cualquier tipo de dato de entrada. Antes de continuar con las funciones propiamente de análisis de los Langleys describiré la alternativa a langley\_summ\_sync.m, es decir, la función langley\_data\_cell.m.

En este caso se introduce como input las variables ***ozone\_raw, ozone\_ds*** y ***config*** (salidas de readb\_ds\_develop.m / readb\_data.m), y se obtienen como salida las siguientes variables:

* ***ozone\_lgl***: serían las medidas individuales (5 por cada sumario, en principio). Los campos disponibles son los dados en la tabla 2.
* ***ozone\_lgl\_sum***: serían los sumarios recalculados a partir de las medidas individuales anteriores. Los campos disponibles son también los dados en la tabla 2. Sin embargo, notar que los sumarios aquí definidos no han sido filtrados según el esquema normal O3\_std < 2.5, 100<O3<600, hg flag =1, n = 5. Es por eso por lo que se añaden campos adicionales 40 (O3(cfg1) dev. std.), 41 (O3(cfg2) dev. std.) y 42 (No. de medidas individuales de cada sumario). La depuración se hará en una fase posterior (**¿Tiene esto sentido? ¿No será mejor depurarlos desde que se producen?**).
* ***cfg***: Constantes de calibración en juego. Estructura con campos old y new.

|  |
| --- |
| ***'Usage date', 'o3 Temp coef 1', 'o3 Temp coef 2', 'o3 Temp coef 3', 'o3 Temp coef 4', 'o3 Temp coef 5', 'O3 on O3 Ratio', 'ETC on O3 Ratio', 'Dead time (sec)', 'ND filter 0', 'ND filter 1', 'ND filter 2', 'ND filter 3', 'ND filter 4', 'ND filter 5'*** |

* ***lgl\_leg***: legenda para cada una de las variables de salida

Paso a comentar a continuación primero la depuración de los datos y luego el análisis de los Langleys:

1. **Depuración de los datos, nivel 1 (langley\_filter\_lvl1.m)**

La función langley\_filter\_lvl1.m recibe como input los datos de langley, bien ***langsumm\_sync\_data*** (datos simultáneos obtenidos a partir de los sumarios) bien ***ozone\_lgl*** (medidas individuales). Funciona con ambas. Los input opcionales se refieren a la depuración que queramos llevar a cabo, que por otra parte se aplica a “medios-días”, AM-PM, definidos a partir del tiempo solar verdadero. Son los siguientes (notar que esto se corresponde con lo que ya se definió en su día como Data Selection Level1):

* airmass: rango de masas ópticas de trabajo. [1.15 3.5] por defecto.
* N\_filt: número mínimo de medidas por filtro. 5 por defecto (NOT IMPLEMENTED).
* N\_hday: número mínimo de medidas para cada “medio-día”. 25 sumarios → 25 x 5 medidas individuales, por defecto.
* O3\_hday: máxima desviación estándar para el ozono promedio de cada “medio-día”. Se pretende con este filtro descartar los “medios-días” en los que el ozono no permanece aproximadamente constante. Por defecto no se usa este filtro.

Existe además un input adicional que, aunque opcional, es importante definir explícitamente, o al menos tener presente su significado. Se trata de ‘summ’. Como hemos dicho, la función langley\_filter\_lvl1.m admite como entrada bien los datos de langley obtenidos a partir de los sumarios (medidas simultáneas), o bien los datos de langley obtenidos a partir de las medidas individuales (función langley\_data\_cell.m). En este caso, se ofrecen dos opciones posibles: 1) medidas individuales y 2) sumarios calculados a partir de esas medidas individuales. Existe entonces un filtro adicional (digamos level 0) que no es más que el fitro rutinario aplicado a los sumarios, es decir O3\_std < 2.5, 100<O3<600, hg flag =1, n = 5. En el caso de que estemos trabajando con la variable ***langsumm\_sync\_data*** entonces este filtro inicial no es necesario, ya ha sido implementado, pero en el caso de que trabajemos bien con la variable ***ozone\_lgl*** bien con ***ozone\_lgl\_sum***, entonces debemos aplicarlos. Si se trata de la primera (***ozone\_lgl***), asignamos a ‘summ’ el valor 0, y lo que obtenemos son las medidas individuales pero se han descartado aquellas cuyo sumario asociado no cumpla con las condiciones anteriores. Si se trata de la segunda (***ozone\_lgl\_sum***) entonces asignamos a ‘summ’ el valor 1, y se aplicará el filtro de los sumarios. En el caso de la variable ***langsumm\_sync\_data*** también debemos hacer ‘summ ‘ = 1, sólo que ahora no pasará nada. En definitiva, siempre que trabajamos con sumarios debemos iniciar el valor del argumento opcional ‘summ’ a 1.

Como salida, esta función devuelve una matriz de datos con igual estructura que la de entrada, pero depurada según los criterios anteriores. Además, en el caso de que asignemos un valor al filtro **O3\_hday** nos devuelve en pantalla una tabla con los “medios-días” aceptados o descartados. Abajo se muestra un ejemplo mostrando el resultado de esta función.

General_.tif

Figure 2: Resultado de la depuración de los datos según los siguiente parámetros: airmass = [1.4 6], N\_hday =25, O3\_hday = 2.5. Los círculos representan datos sin filtrar de ninguna manera.

Tabla 3: Parámetros asociados a cada día del análisis anterior, según los parámetros de filtrado airmass = [1.4 6], N\_hday =25, O3\_hday = 2.5.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Diaj** | **AM** | **PM** | **std\_am** | **N\_am** | **std\_pm** | **N\_pm** |
| **15-Apr-2013** | 105 | 1 | 0 | 2.30 | 28.0 | 1.35 | 23.0 |
| **16-Apr-2013** | 106 | 0 | 0 | 3.32 | 28.0 | 3.86 | 25.0 |
| **17-Apr-2013** | 107 | 1 | 1 | 1.41 | 28.0 | 1.61 | 25.0 |
| **18-Apr-2013** | 108 | 0 | 1 | 2.52 | 5.0 | 1.95 | 25.0 |
| **19-Apr-2013** | 109 | 0 | 0 | 1.11 | 21.0 | 0.83 | 2.0 |
| **20-Apr-2013** | 110 | 1 | 0 | 1.34 | 27.0 | 1.91 | 16.0 |
| **21-Apr-2013** | 111 | 1 | 0 | 1.26 | 28.0 | 3.23 | 17.0 |
| **22-Apr-2013** | 112 | 1 | 1 | 0.71 | 28.0 | 1.41 | 25.0 |
| **23-Apr-2013** | 113 | 0 | 0 | 2.55 | 28.0 | 1.11 | 23.0 |
| **24-Apr-2013** | 114 | 1 | 0 | 1.72 | 27.0 | 0.63 | 5.0 |
| **25-Apr-2013** | 115 | 0 | 0 | 1.26 | 24.0 | 3.58 | 18.0 |
| **26-Apr-2013** | 116 | 1 | 0 | 1.67 | 27.0 | 2.38 | 16.0 |
| **27-Apr-2013** | 117 | 0 | 1 | 1.36 | 20.0 | 1.94 | 26.0 |
| **28-Apr-2013** | 118 | 0 | 0 | 1.07 | 24.0 | 1.20 | 7.0 |
| **29-Apr-2013** | 119 | 0 | 1 | 2.57 | 27.0 | 1.93 | 29.0 |
| **30-Apr-2013** | 120 | 1 | 1 | 2.13 | 25.0 | 0.99 | 29.0 |

1. **Obtención de la constante ETC mediante Langley plot (langley\_analys.m)**

La función langley\_analys.m realiza una regresión simple de MS9 frente a la masa óptica para obtener la constante ETC. Aunque no está completa, los cálculos básicos si están implementados.

Los argumentos de entrada son, como obligatorios, los siguientes:

1. **lgl\_data:** puede ser o bien la salida depurada de la función langley\_filter\_lvl1.m o bien cualquiera de las variables (no depuradas) ***langsumm\_sync\_data,*** ***ozone\_lgl*** ó ***ozone\_lgl\_sum.***
2. **brw:** puesto que los datos de entrada van a ser cell-arrays, uno para cada instrumento, con este parámetro identificamos el instrumento que nos interesa estudiar.
3. **Cal:** esta es una variable-estructura normalmente resultado de la evaluación del fichero de definiciones (por ejemplo calizo2013\_setup.m). En todo caso de aquí sólo se toman las configuraciones 1 y 2.

|  |
| --- |
| General_.tif |
| General_.tif |

Figure 3: Efecto del filtrado adicional ‘res\_filt’. Se eliminan los datos cuyo residuo sea mayor que 1.5 veces la desviación estándar del total de residuos. Datos brutos (no se ha aplicado ningún otro filtro). Método Brewer (arriba) y método Dobson (abajo).

Además de los argumentos obligatorios anteriores, existe uno opcional ***‘res\_filt’.*** Este parámetro controla si se realiza o no un filtro adicional de depuración sobre los datos de la regresión, descartando aquellos que verifican la condición abs(residuos)>1.5 x std(residuos). Es decir, se eliminan aquellos datos cuyo residuo sea mayor que 1.5 veces la desviación estándar del total ([Michalsky](http://www.esrl.noaa.gov/gmd/grad/neubrew/docs/publications/Harrison_ApplOpt1994.pdf)). Una vez hecho esto, se vuelve a calcular la regresión con los nuevos datos depurados.

Como posibles salidas, hasta ahora la función devuelve las variables ***resp\_brw*** y ***resp\_dbs.*** En ambos casos ***s***e trata de matrices 3-D (:,3,2), donde:

* 1D: una fila para cada día analizado (con al menos AM o PM válidos)
* 2D: 3 columnas. Date, Langley AM y Langley PM
* 3D: 1 config / 2 config

Optional

  a) Data selected michalsky

<http://www.esrl.noaa.gov/gmd/grad/neubrew/docs/publications/Harrison_ApplOpt1994.pdf>

    same output but adding  a flag

  b) AOD from aeronet from the same period

  c) Color index

  d) Cloud observations / Cloud camera

 2) Data selection 1

    “Variables”  for selection  level 1

           b) n air mass range  -> default 1.15-3.5

           c) n observations  ( equally distributed better) -> default 25

           d) number of measurements by filter                 -> default 5

           e) Filter correction                                              -> default from config

3)    Regression  (MS9-ETCR)/n  vs 1/n ->  Linregress -> Store all the outputs of linregress  !!. In addition:  airmass range (min, max) , n obs

 output 1;  daily (AM/PM) langleyplot  (1/nu)  am/pm of the langley colors by filter  by flag

                 R6 vs time

Summary output 1:

**Simultaneous ETC correction for all the brewer per day per config**

**ETC  correction instrument vs instrument**

**ETC  correction instrument vs AOD**

4) Recalculation

        ozone mean and std recalculated with ETC corrected

5)  Filter #2  for regression results

                  n obs

                  ozone mean and std during langley

                  R coeff corr

                  Error in slope