SA-180612 - All Figures with Uncertainties

August 30, 2018

```
In [1]: %pylab inline

Populating the interactive namespace from numpy and matplotlib


In [2]: from SA_import import *

loading science.mplstyle


In [3]: foster_data_url="https://media.nature.com/original/nature-
        assets/ncomms/2017/170404/ncomms14845/extref/ncomms14845-s3.xlsx"
        foster_data_file='ncomms14845-s3.xlsx'

        download(foster_data_url,foster_data_file,overwrite=False)

File  ncomms14845-s3.xlsx exists...not overwriting


Out[3]: False

In [4]: errorbar_level=68

In [5]: saveit=True

In [6]: %%time
        r=get_results()

CPU times: user 16.4 s, sys: 1.41 s, total: 17.8 s
Wall time: 17.9 s
```

0.1 Export the results

```
In [7]: num_data=len(r['Ma'])

In [8]: keys=['Ma', 'Ma_L','Ma_U', 'Ph', '', 'p', 'a', 'TC', 'TK', 'b', 'd', 'p', 'S', 'KO', 'b', 'f',
              'dissolved_CO2', 'pCO2', 'current_oil','current_sedimentary','other_oil','other_sedimentary']

In [9]: for c in ['current_oil','current_sedimentary','other_oil','other_sedimentary']:
            r[c]=zeros(num_data)
            r[c][r['index_'+c]]=1

In [10]: data={}
         new_keys=[]
         for key in keys:
             vals=r[key]
             if vals.shape==(num_data,):
                 data[key]=vals
                 new_keys.append(key)
             elif vals.shape[0]==num_data:
                 y=np.median(vals,axis=1)
                 yl=np.percentile(vals,(100-68)/2,axis=1)
                 yu=np.percentile(vals,100-(100-68)/2,axis=1)
                 data[key]=y
                 new_keys.append(key)
```

```
            data[key+"_L1"]=yl
            new_keys.append(key+"_L1")
            data[key+"_U1"]=yu
            new_keys.append(key+"_U1")

            yl=np.percentile(vals,(100-95)/2,axis=1)
            yu=np.percentile(vals,100-(100-95)/2,axis=1)
            data[key+"_L2"]=yl
            new_keys.append(key+"_L2")
            data[key+"_U2"]=yu
            new_keys.append(key+"_U2")

        else:
            raise ValueError(key)
```

In [11]:
```
data=pandas.DataFrame(data)
data=data[new_keys]
```

In [12]:
```
data.head()
```

Out[12]:

	Ma	Ma_L	Ma_U	Ph	Ph_L1	Ph_U1	Ph_L2	Ph_U2	\
0	0.1	0.3	0.0	-25.260302	-25.599151	-24.919090	-25.734962	-24.784824	
1	0.5	1.0	0.0	-24.376151	-24.717525	-24.040780	-24.855048	-23.905646	
2	0.8	1.0	0.5	-25.729731	-26.071187	-25.389393	-26.204843	-25.255360	
3	1.8	2.0	1.5	-24.102529	-24.438695	-23.761691	-24.573481	-23.625812	
4	2.1	2.3	2.0	-24.297099	-24.641087	-23.960279	-24.775674	-23.825200	

	_L1	...	dissolved_CO2_U2	pCO2	\
0	3.491109	2.208745	...	16.983969	282.539456
1	3.510182	2.226826	...	15.548633	247.983095
2	3.513479	2.238523	...	17.488370	282.916902
3	3.515069	2.230297	...	15.993373	274.516389
4	3.513783	2.244820	...	17.128712	306.787236

	pCO2_L1	pCO2_U1	pCO2_L2	pCO2_U2	current_oil	\
0	209.179136	372.472331	160.425607	469.673198	0.0	
1	182.901011	326.618213	141.260622	408.583055	0.0	
2	208.232668	373.983185	160.519823	473.774235	0.0	
3	204.039045	361.177317	158.346190	454.054166	0.0	
4	227.404406	404.300006	176.564796	509.383088	0.0	

	current_sedimentary	other_oil	other_sedimentary
0	1.0	0.0	0.0
1	1.0	0.0	0.0
2	1.0	0.0	0.0
3	1.0	0.0	0.0
4	1.0	0.0	0.0

[5 rows x 78 columns]

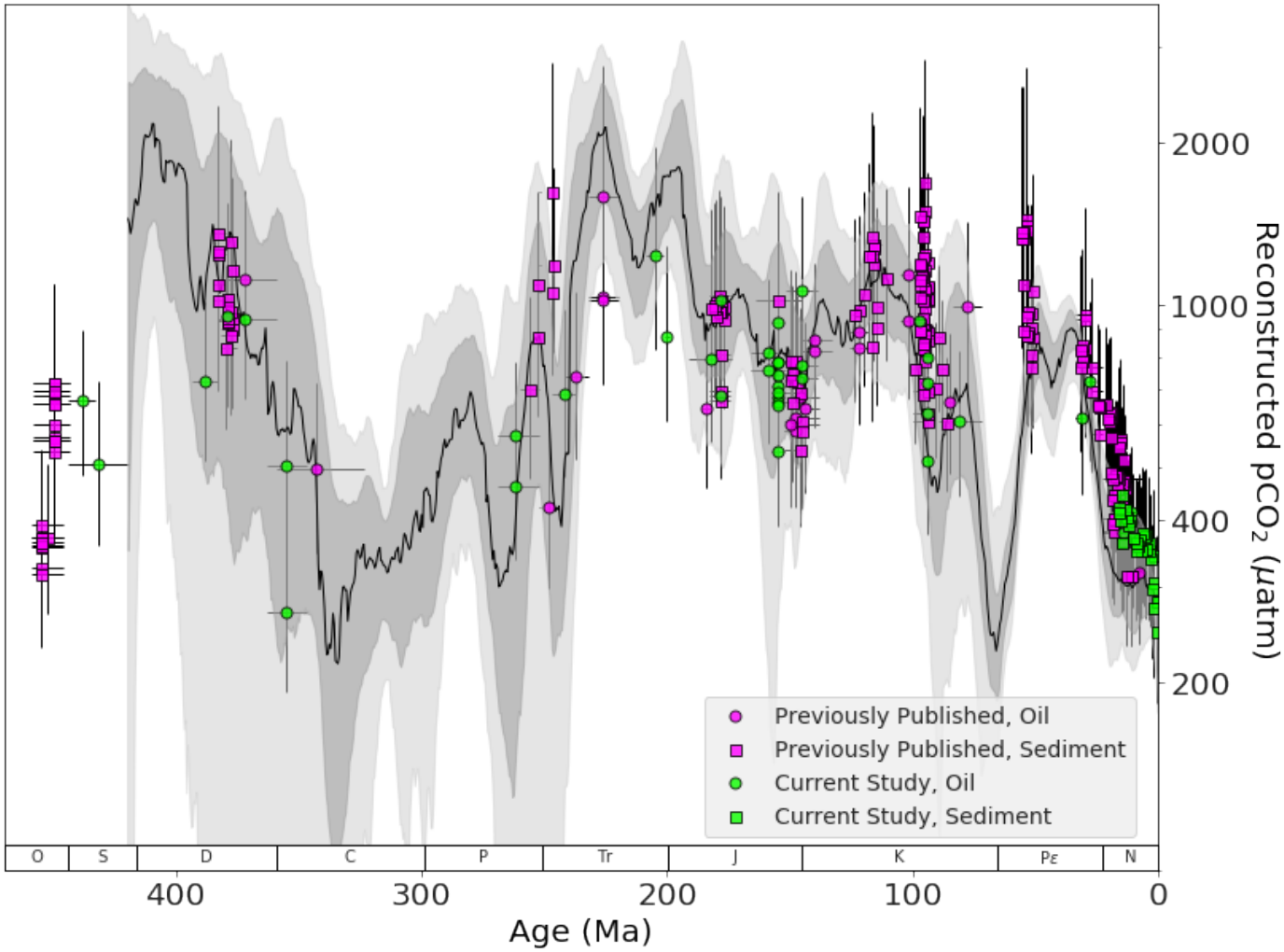
In [13]:
```
data.to_excel('SA_180612 export.xlsx')
```

0.2 *Figure pCO_2 with age error bars and with the uncertainty and Foster Data*

In [14]:
```
color1='magenta'
#color2='#61cfe2'  # cyan
color2='#FF9C33'  # orange
#color2='y'
color2='#0EFF00'
```

In [15]:
```
figure(figsize=(13,10))

lwx=r['Foster']['lwx']
lwy=r['Foster']['lwy']
lw68=r['Foster']['lw68']
lw95=r['Foster']['lw95']
up68=r['Foster']['up68']
up95=r['Foster']['up95']

plot(lwx,lwy,'k-',lw=1)
fill_between(lwx, lw95, up95,
    alpha=0.5, edgecolor='#CCCCCC', facecolor='#CCCCCC')
fill_between(lwx, lw68, up68,
    alpha=0.5, edgecolor='#999999', facecolor='#999999')


#==========


x=array(r['Ma'])
xu=array(r['Ma_U'])
xl=array(r['Ma_L'])

idx_os_mo=r['index_other_oil']
idx_os_ms=r['index_other_sedimentary']
idx_cs_mo=r['index_current_oil']
idx_cs_ms=r['index_current_sedimentary']

y=np.median(r['pCO2'],axis=1)
yl=np.percentile(r['pCO2'],(100-errorbar_level)/2,axis=1)
yu=np.percentile(r['pCO2'],100-(100-errorbar_level)/2,axis=1)


xerr=array([x-xl,xu-x])
yerr=array([y-yl,yu-y])

alpha=0.8

plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
    label='Previously Published, Oil',alpha=alpha)
plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
    label='Previously Published, Sediment',alpha=alpha)
plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
    label='Current Study, Oil',alpha=alpha)
plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
    label='Current Study, Sediment',alpha=alpha)

errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
ax=gca()
ax.invert_xaxis()
xlim([470,0])

xlabel('Age (Ma)')

ax.set_yscale('log')
h=ylabel('Reconstructed pCO$_2$ ($\mu$atm)',rotation=270,va='bottom', labelpad=20)

ylim([90,3600])

ax.yaxis.set_label_position("right")
ax.yaxis.set_ticks_position("right")
ax.set_yticks([200,400,1000,2000])
ax.set_yticklabels([200,400,1000,2000])
grid('off')

x2,y2,xerr2,yerr2=x,y,xerr,yerr

add_time_periods(90,100,10)
LL=legend(loc='lower right',frameon=True,fontsize=14,
              bbox_to_anchor=(0.99, 0.025))
LL.get_frame().set_facecolor('#EEEEEE')

#==========

if saveit:
```

```
plt.savefig("../figures/Fig pCO2 with age error bars overlaid on the Foster.png",dpi=900,bbox_inches='tight')
```

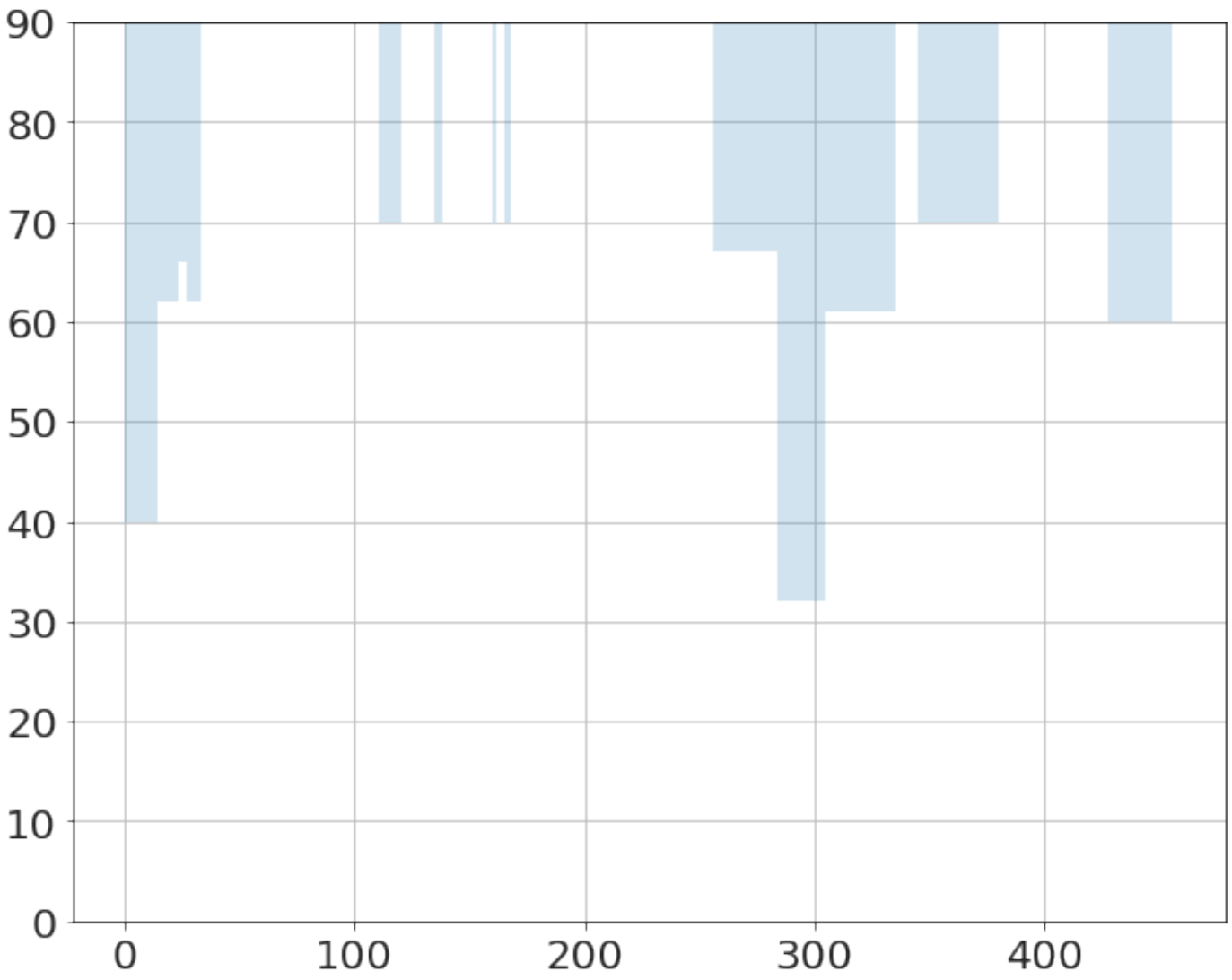


0.3 same as fig above, but with glaciation

In [16]:
```
data_str="""
Start	Stop	ž
456	428	60
380	345	70
335	305	61
305	284	32
284	256	67
168	165	70
162	160	70
138	135	70
120	110	70
33.5	27	62
27	23.5	66
23.5	23	76
23	14	62
14	0	40
"""
lines=data_str.strip().split('\n')
```

In [17]:
```
center=[]
width=[]
height=[]
for line in lines[1:]:
    parts=line.split('\t')
```

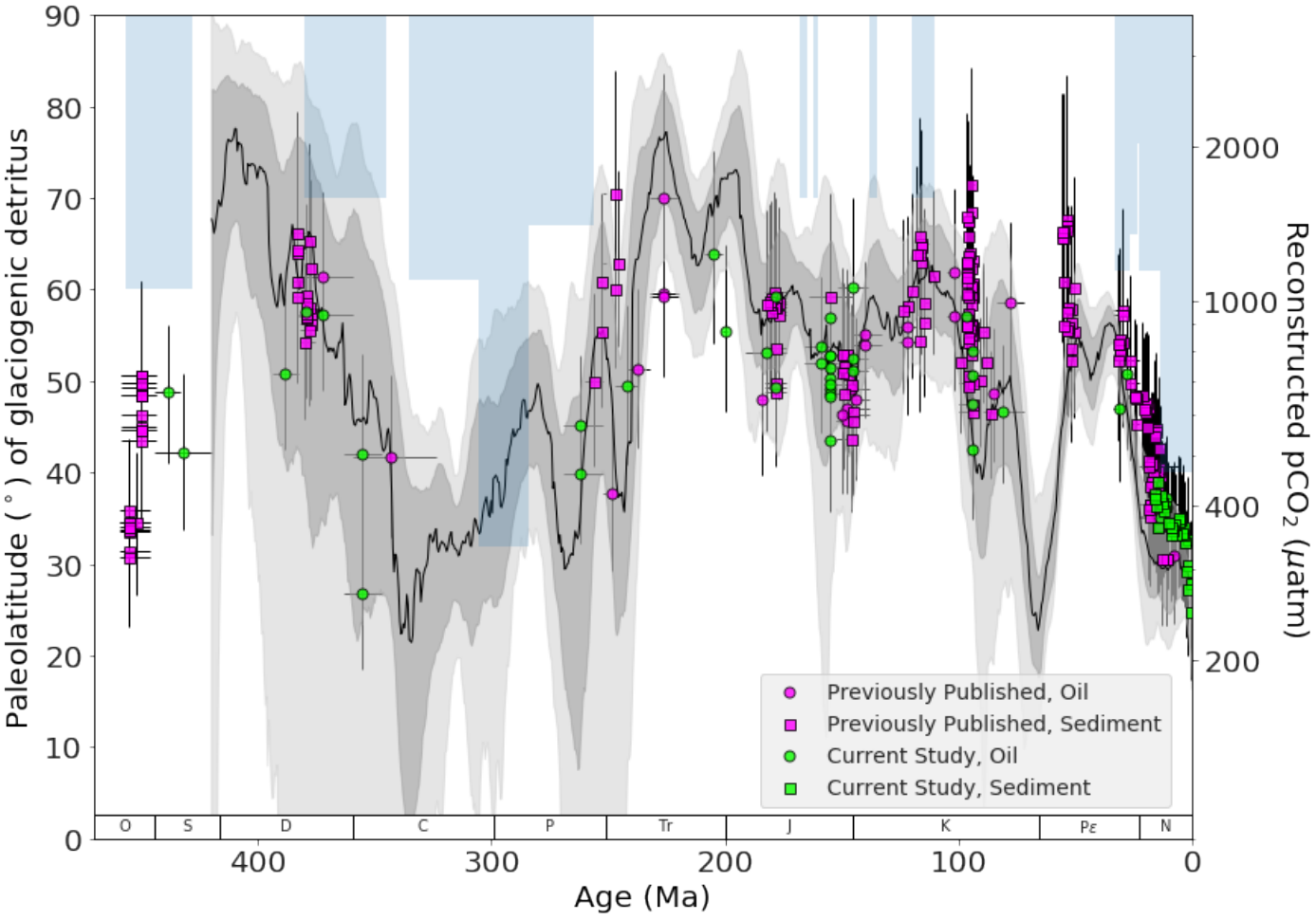
```
    center.append((float(parts[0])+float(parts[1]))/2)
    width.append((float(parts[1])-float(parts[0])))
    height.append(90-float(parts[2]))
bar(center,height,width,alpha=0.2)
gca().set_ylim([0,90])
yt=list(range(0,91,10))
gca().set_yticks(yt)
yt.reverse()
gca().set_yticklabels(yt)
gca().invert_yaxis()
```



In [18]:
```
figure(figsize=(13,10))

lwx=r['Foster']['lwx']
lwy=r['Foster']['lwy']
lw68=r['Foster']['lw68']
lw95=r['Foster']['lw95']
up68=r['Foster']['up68']
up95=r['Foster']['up95']

plot(lwx,lwy,'k-',lw=1)
fill_between(lwx, lw95, up95,
    alpha=0.5, edgecolor='#CCCCCC', facecolor='#CCCCCC')
fill_between(lwx, lw68, up68,
    alpha=0.5, edgecolor='#999999', facecolor='#999999')
```

```
#=========


x=array(r['Ma'])
xu=array(r['Ma_U'])
xl=array(r['Ma_L'])

idx_os_mo=r['index_other_oil']
idx_os_ms=r['index_other_sedimentary']
idx_cs_mo=r['index_current_oil']
idx_cs_ms=r['index_current_sedimentary']

y=np.median(r['pCO2'],axis=1)
yl=np.percentile(r['pCO2'],(100-errorbar_level)/2,axis=1)
yu=np.percentile(r['pCO2'],100-(100-errorbar_level)/2,axis=1)


xerr=array([x-xl,xu-x])
yerr=array([y-yl,yu-y])

alpha=0.8


ax=gca()
ax2=ax.twinx()
ax2.bar(center,height,width,alpha=0.2,label='glaciogenic latitude')
ax2.set_ylim([0,90])
yt=list(range(0,91,10))
ax2.set_yticks(yt)
yt.reverse()
ax2.set_yticklabels(yt)
ax2.invert_yaxis()
ax2.yaxis.set_label_position("left")
ax2.yaxis.set_ticks_position("left")
ax2.set_ylabel('Paleolatitude ($^\circ$) of glaciogenic detritus')
ax2.grid('off')

ax.plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
    label='Previously Published, Oil',alpha=alpha)
ax.plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
    label='Previously Published, Sediment',alpha=alpha)
ax.plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
    label='Current Study, Oil',alpha=alpha)
ax.plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
    label='Current Study, Sediment',alpha=alpha)

ax.errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
#ax=gca()
ax.invert_xaxis()
ax.set_xlim([470,0])

ax.set_xlabel('Age (Ma)')

ax.set_yscale('log')
h=ax.set_ylabel('Reconstructed pCO$_2$ ($\mu$atm)',rotation=270,va='bottom', labelpad=20)

ax.set_ylim([90,3600])

ax.yaxis.set_label_position("right")
ax.yaxis.set_ticks_position("right")
ax.set_yticks([200,400,1000,2000])
ax.set_yticklabels([200,400,1000,2000])
ax.grid('off')

x2,y2,xerr2,yerr2=x,y,xerr,yerr

add_time_periods(90,100,10,ax=ax)
LL=ax.legend(loc='lower right',frameon=True,fontsize=14,
             bbox_to_anchor=(0.99, 0.025))
LL.get_frame().set_facecolor('#EEEEEE')



#=========
```

```
if saveit:
    plt.savefig("../figures/Fig pCO2 with age error bars overlaid on the Foster with
glaciation.png",dpi=900,bbox_inches='tight')
```

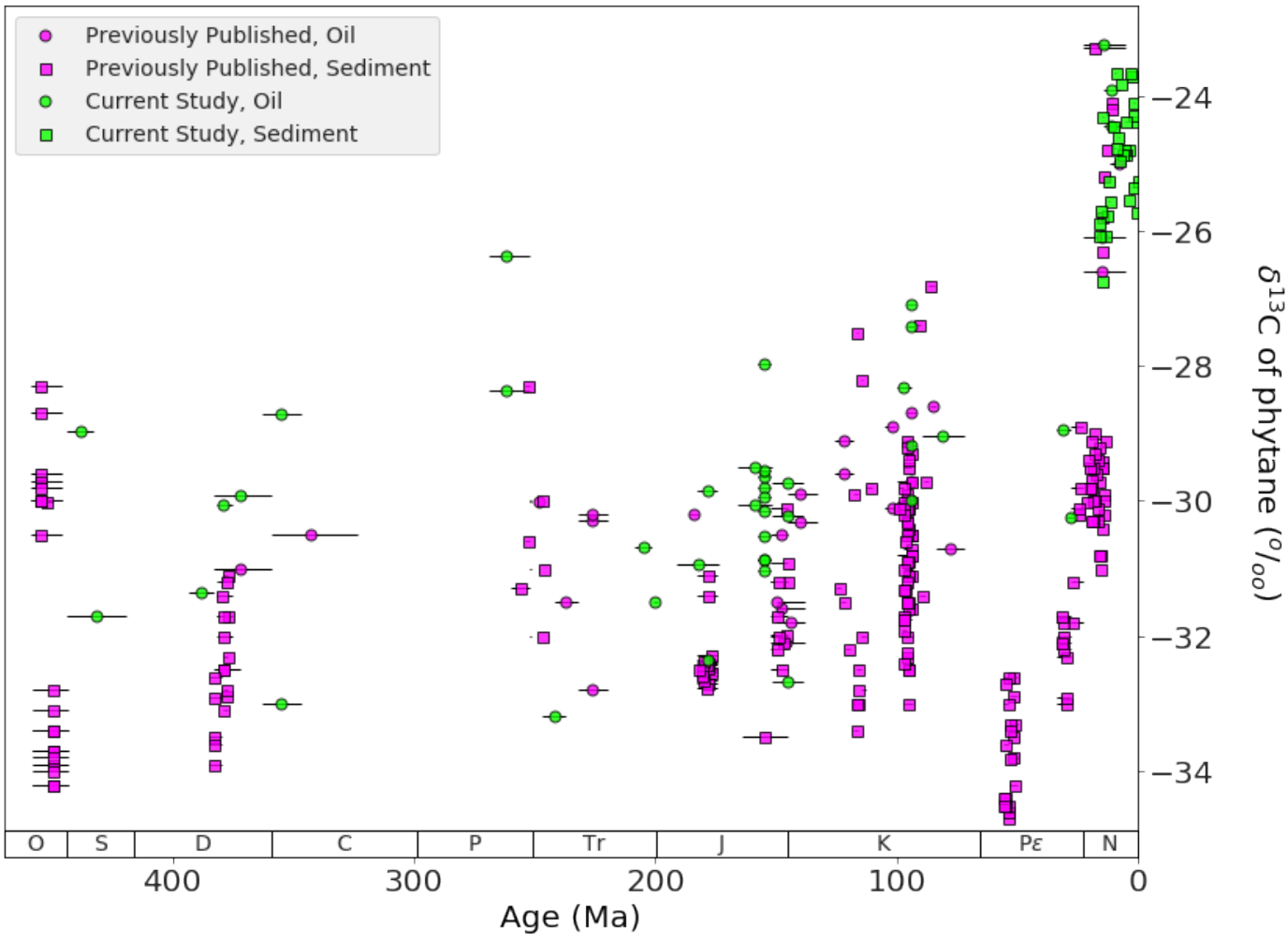


1 *Figure ^{13}C of phytane with age error bars*

In [19]: `figure(figsize=(13,10))`

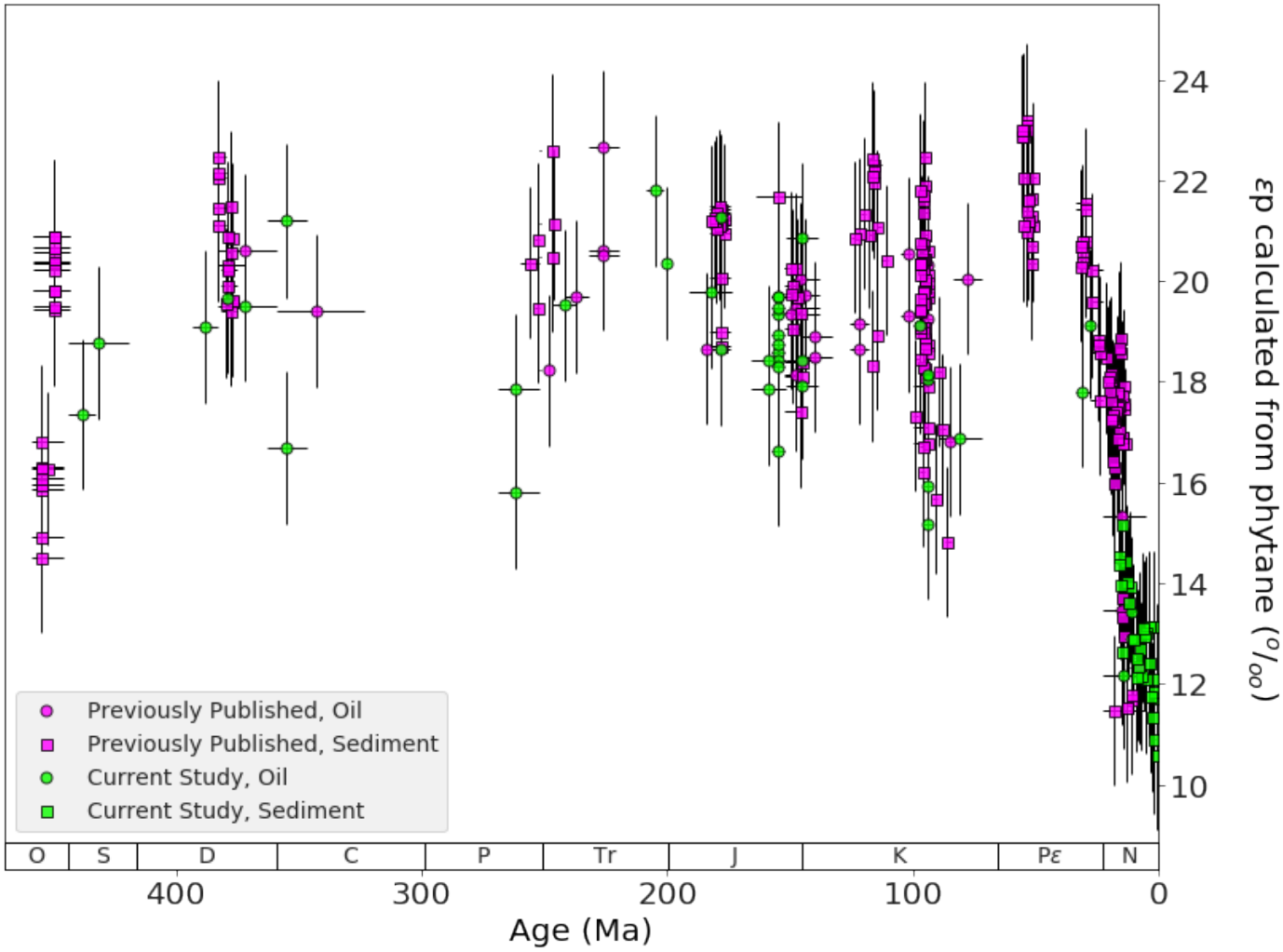
```
x=array(r['Ma'])
xu=array(r['Ma_U'])
xl=array(r['Ma_L'])

idx_os_mo=r['index_other_oil']
idx_os_ms=r['index_other_sedimentary']
idx_cs_mo=r['index_current_oil']
idx_cs_ms=r['index_current_sedimentary']

y=np.median(r['Ph'],axis=1)
yl=np.percentile(r['Ph'],(100-errorbar_level)/2,axis=1)
yu=np.percentile(r['Ph'],100-(100-errorbar_level)/2,axis=1)


xerr=array([x-xl,xu-x])
yerr=array([y-yl,yu-y])

alpha=0.8
plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
```

```
    label='Previously Published, Oil',alpha=alpha)
plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
    label='Previously Published, Sediment',alpha=alpha)
plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
    label='Current Study, Oil',alpha=alpha)
plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
    label='Current Study, Sediment',alpha=alpha)

errorbar(x,y,xerr=xerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
ax=gca()
ax.invert_xaxis()
xlim([470,0])
xlabel('Age (Ma)')


ax.yaxis.set_label_position("right")
ax.yaxis.set_ticks_position("right")

h=ylabel(r'$\delta^{13}$C of phytane ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
#h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%}$)',rotation=270,va='bottom', labelpad=20)

grid('off')

add_time_periods()
LL=legend(loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')

#=========

if saveit:
    plt.savefig("../figures/Fig 13C of phytane with age error bars.png",dpi=900,bbox_inches='tight')
```



2 Figure p of phytane with age error bars

In [20]:
```
figure(figsize=(13,10))


        x=array(r['Ma'])
        xu=array(r['Ma_U'])
        xl=array(r['Ma_L'])

        idx_os_mo=r['index_other_oil']
        idx_os_ms=r['index_other_sedimentary']
        idx_cs_mo=r['index_current_oil']
        idx_cs_ms=r['index_current_sedimentary']

        y=np.median(r['p'],axis=1)
        yl=np.percentile(r['p'],(100-errorbar_level)/2,axis=1)
        yu=np.percentile(r['p'],100-(100-errorbar_level)/2,axis=1)


        xerr=array([x-xl,xu-x])
        yerr=array([y-yl,yu-y])

        alpha=0.8
        plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Oil',alpha=alpha)
        plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Sediment',alpha=alpha)
        plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Oil',alpha=alpha)
        plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Sediment',alpha=alpha)



        errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
        ax=gca()
        ax.invert_xaxis()
        xlim([470,0])
        xlabel('Age (Ma)')


        ax.yaxis.set_label_position("right")
        ax.yaxis.set_ticks_position("right")

        h=ylabel(r'$\epsilon$p calculated from phytane ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
        #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

        grid('off')

        add_time_periods()
        LL=legend(loc='lower left',frameon=True,fontsize=14,bbox_to_anchor=(0,.03))
        LL.get_frame().set_facecolor('#EEEEEE')

        #=========


        if saveit:
            plt.savefig("../figures/Fig p of phytane with age error bars and uncertainty.png",dpi=900,bbox_inches='tight')
```



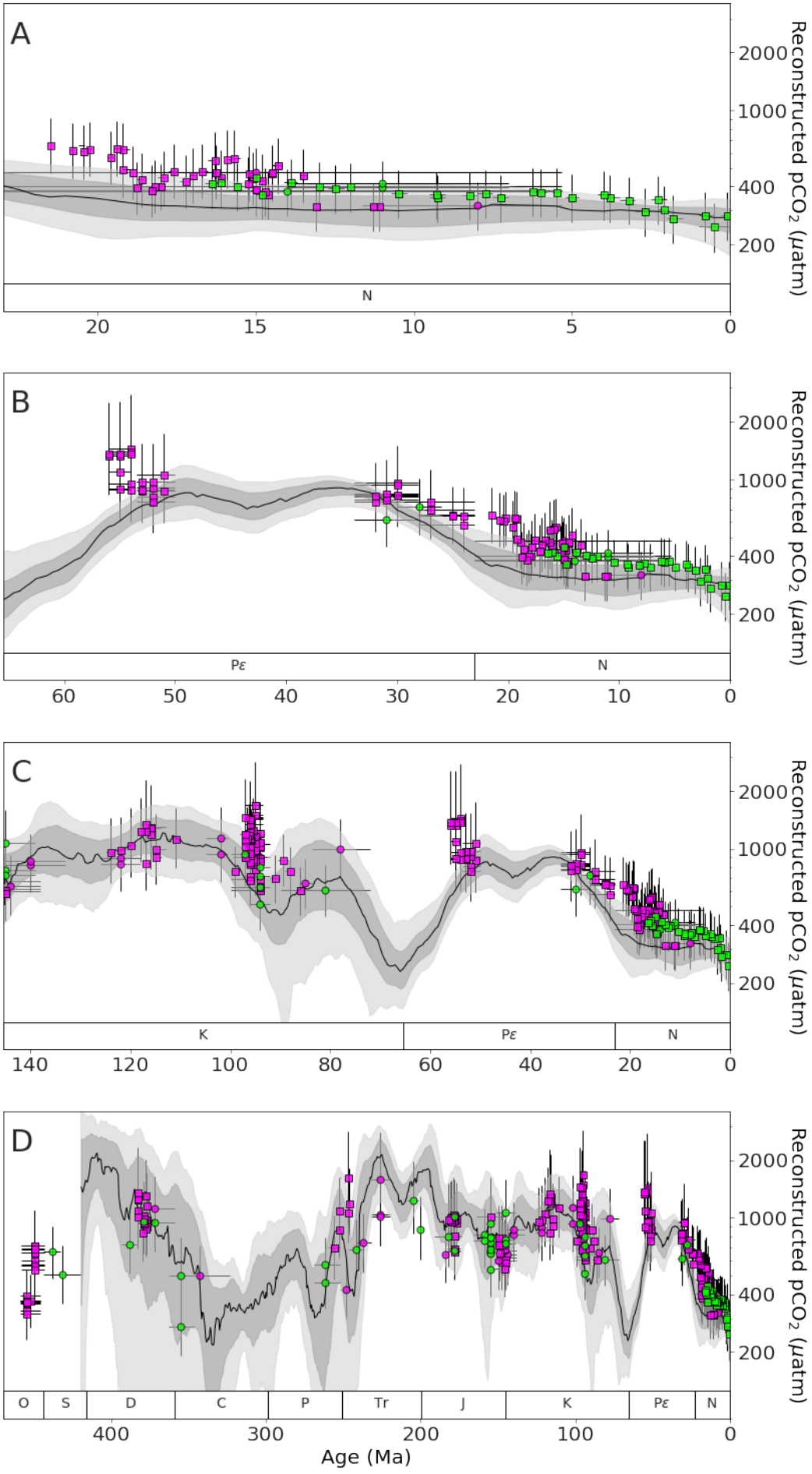
3 Figure pCO2 reconstruction with Foster et al., zoomed in for specific time periods:

A: Phanerozoic (what we currently have)
B: Cretaceous to today
C: Paleogene to today
D: Neogene to today

```
In [21]: figure(figsize=(13,26))

         lwx=r['Foster']['lwx']
         lwy=r['Foster']['lwy']
         lw68=r['Foster']['lw68']
         lw95=r['Foster']['lw95']
         up68=r['Foster']['up68']
         up95=r['Foster']['up95']


         #==========


         x=array(r['Ma'])
         xu=array(r['Ma_U'])
         xl=array(r['Ma_L'])

         idx_os_mo=r['index_other_oil']
         idx_os_ms=r['index_other_sedimentary']
```

```
idx_cs_mo=r['index_current_oil']
idx_cs_ms=r['index_current_sedimentary']

y=np.median(r['pCO2'],axis=1)
yl=np.percentile(r['pCO2'],(100-errorbar_level)/2,axis=1)
yu=np.percentile(r['pCO2'],100-(100-errorbar_level)/2,axis=1)


xerr=array([x-xl,xu-x])
yerr=array([y-yl,yu-y])


def plotit(xl=[470,0],frac=0.03):
    plot(lwx,lwy,'k-',lw=1)
    fill_between(lwx, lw95, up95,
        alpha=0.5, edgecolor='#CCCCCC', facecolor='#CCCCCC')
    fill_between(lwx, lw68, up68,
        alpha=0.5, edgecolor='#999999', facecolor='#999999')



    alpha=0.8

    plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Oil',alpha=alpha)
    plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Sediment',alpha=alpha)
    plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Oil',alpha=alpha)
    plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Sediment',alpha=alpha)

    errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
    ax=gca()
    ax.invert_xaxis()
    xlim(xl)


    xlabel('Age (Ma)')

    ax.set_yscale('log')
    h=ylabel('Reconstructed pCO$_2$ ($\mu$atm)',rotation=270,va='bottom', labelpad=20)

    ylim([90,3600])

    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")
    ax.set_yticks([200,400,1000,2000])
    ax.set_yticklabels([200,400,1000,2000])
    grid('off')

    x2,y2,xerr2,yerr2=x,y,xerr,yerr

    add_time_periods(frac=frac)

#==========
def letter_label(letter):
    text(0.01, 0.85,letter,fontsize=30,
         horizontalalignment='left',
         verticalalignment='bottom',
         transform = gca().transAxes)



subplot(4,1,1)
# Neogene to today
plotit([22.99,0],frac=0.01)
xlabel('')
letter_label('A')

subplot(4,1,2)
# Paleogene to today
plotit([65.49,0],frac=0.01)
xlabel('')
letter_label('B')
```

```
subplot(4,1,3)
# Cretaceous to today
plotit([145.49,0],frac=0.01)
xlabel('')
letter_label('C')

subplot(4,1,4)
plotit(frac=0.01)
# LL=legend(loc='upper left',frameon=True,fontsize=14)
# LL.get_frame().set_facecolor('#EEEEEE')
letter_label('D')


if saveit:
    plt.savefig("../figures/Fig pCO2 with Foster zoomed in for specific time periods.png",dpi=900,bbox_inches='tight')
```

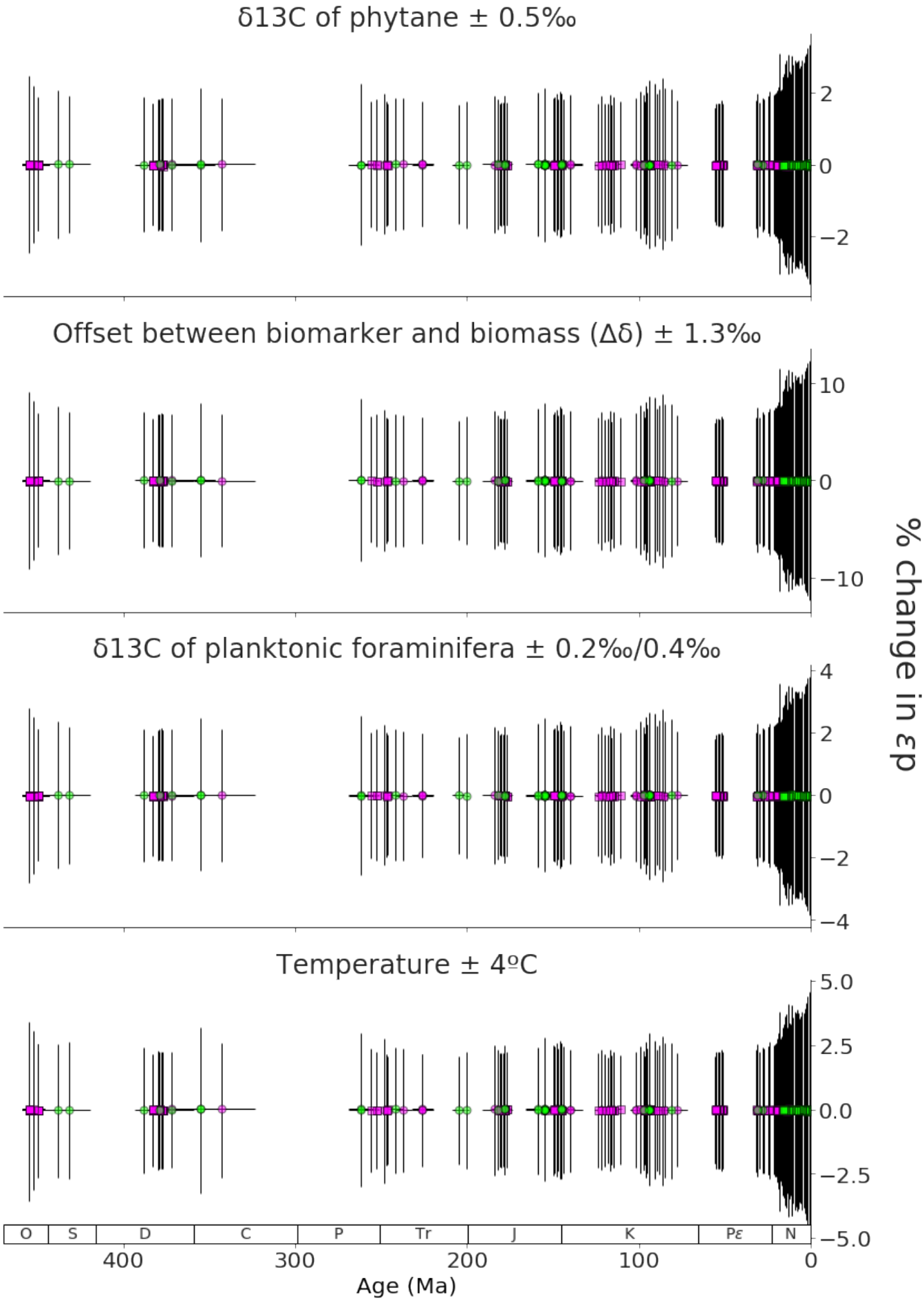

A
B
C
D
Reconstructed pCO2 (μatm)
2000
1000
400
200
N
Pε
K
O
S
D
C
P
Tr
J
20
15
10
5
0
60
50
40
30
140
120
100
80
400
300
200
100
Age (Ma)

4 Fig S3a. p of phytane breakdown on the uncertainty associated with each individual parameter

4.0.1 (everything kept as is, with the exception of the one parameter being highlighted).

```
In [22]: %%time
         r_baseline=get_results(fix_parameters=['Ph','','a','TC'])
         r_Ph=get_results(fix_parameters=['','a','TC'])
         r_=get_results(fix_parameters=['Ph','a','TC'])
         r_a=get_results(fix_parameters=['Ph','','TC'])
         r_TC=get_results(fix_parameters=['Ph','','a'])

CPU times: user 1min 17s, sys: 8.05 s, total: 1min 25s
Wall time: 1min 26s


In [23]: figure(figsize=(13,20))

         def plotit(r,r_baseline,frac=0.03):
             x=array(r['Ma'])
             xu=array(r['Ma_U'])
             xl=array(r['Ma_L'])

             idx_os_mo=r['index_other_oil']
             idx_os_ms=r['index_other_sedimentary']
             idx_cs_mo=r['index_current_oil']
             idx_cs_ms=r['index_current_sedimentary']


             val=(r['p']-r_baseline['p'])/r_baseline['p']*100
             y=np.median(val,axis=1)
             yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
             yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


             xerr=array([x-xl,xu-x])
             yerr=array([y-yl,yu-y])

             alpha=0.5
             plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                 label='Previously Published, Oil',alpha=alpha)
             plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                 label='Previously Published, Sediment',alpha=alpha)
             plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                 label='Current Study, Oil',alpha=alpha)
             plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                 label='Current Study, Sediment',alpha=alpha)



             errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
             ax=gca()
             ax.invert_xaxis()
             xlim([470,0])
             xlabel('Age (Ma)')


             ax.yaxis.set_label_position("right")
             ax.yaxis.set_ticks_position("right")

             #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
             #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

             grid('off')

             add_time_periods(frac=frac)

             #==========
```

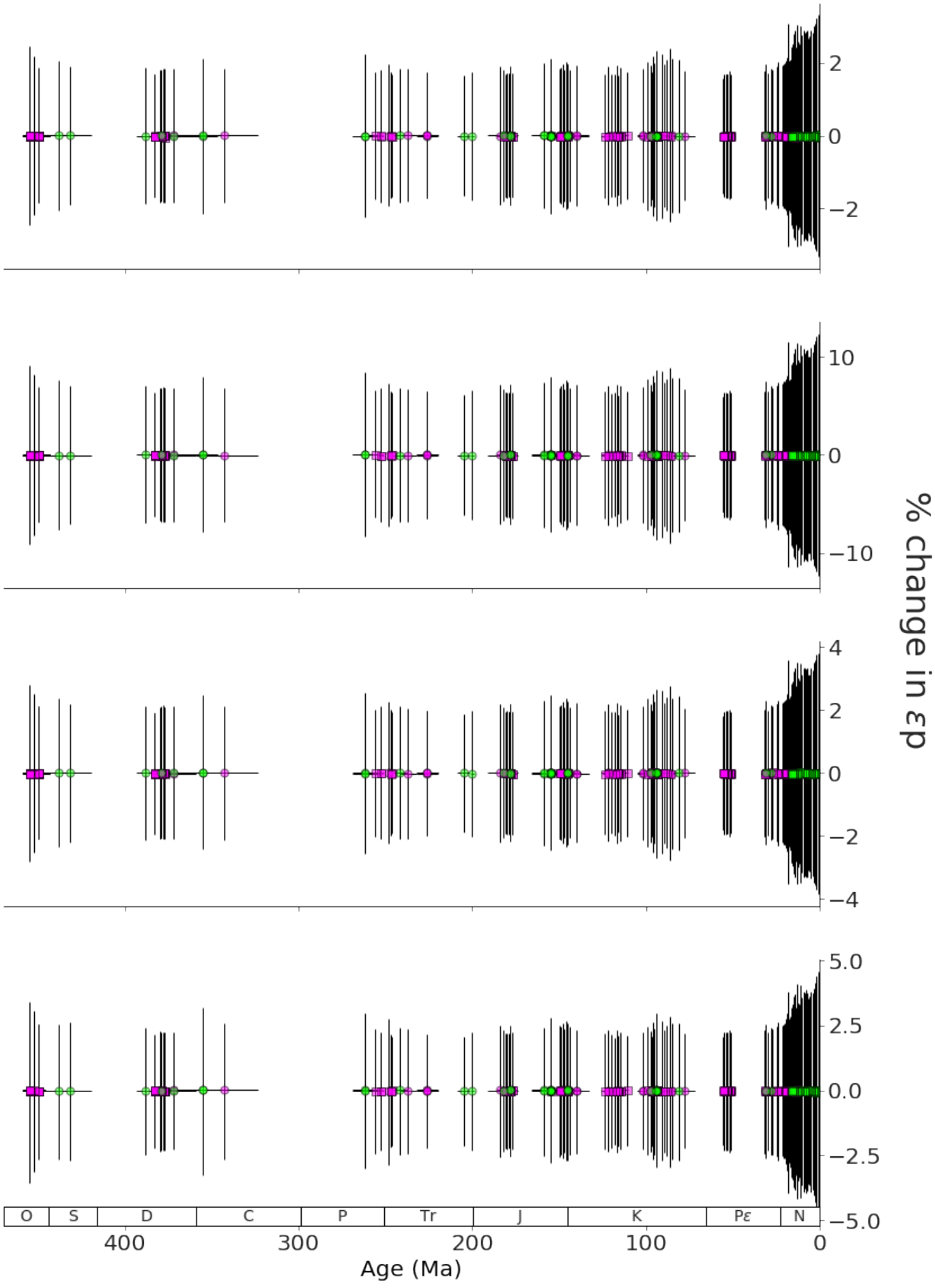
```
subplot(4,1,1)
plotit(r_Ph,r_baseline,frac=0)
title('13C of phytane ś 0.5')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,2)
plotit(r_,r_baseline,frac=0)
title('Offset between biomarker and biomass () ś 1.3')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,3)
plotit(r_a,r_baseline,frac=0)
title('13C of planktonic foraminifera ś 0.2/0.4')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,4)
plotit(r_TC,r_baseline,frac=0.07)
title('Temperature ś 4žC')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

gcf().text(0.99, 0.5, r'% change in $\epsilon$p', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig percent change p uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


δ13C of phytane ± 0.5‰
2
0
−2
Offset between biomarker and biomass (Δδ) ± 1.3‰
10
0
−10
δ13C of planktonic foraminifera ± 0.2‰/0.4‰
4
2
0
−2
−4
Temperature ± 4ºC
5.0
2.5
0.0
−2.5
−5.0
% change in εp
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)

4.0.2 same as above but no titles

In [24]:
```
figure(figsize=(13,20))

    def plotit(r,r_baseline,frac=0.03):
        x=array(r['Ma'])
        xu=array(r['Ma_U'])
        xl=array(r['Ma_L'])

        idx_os_mo=r['index_other_oil']
        idx_os_ms=r['index_other_sedimentary']
        idx_cs_mo=r['index_current_oil']
        idx_cs_ms=r['index_current_sedimentary']


        val=(r['p']-r_baseline['p'])/r_baseline['p']*100
        y=np.median(val,axis=1)
        yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
        yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


        xerr=array([x-xl,xu-x])
        yerr=array([y-yl,yu-y])

        alpha=0.5
        plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Oil',alpha=alpha)
        plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Sediment',alpha=alpha)
        plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Oil',alpha=alpha)
        plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Sediment',alpha=alpha)



        errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
        ax=gca()
        ax.invert_xaxis()
        xlim([470,0])
        xlabel('Age (Ma)')


        ax.yaxis.set_label_position("right")
        ax.yaxis.set_ticks_position("right")

        #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
        #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%}$)',rotation=270,va='bottom', labelpad=20)

        grid('off')

        add_time_periods(frac=frac)

        #==========

    subplot(4,1,1)
    plotit(r_Ph,r_baseline,frac=0)
    #title('13C of phytane ś 0.5')
    gca().set_xticklabels([])
    gca().set_xlabel('')
    gca().spines['left'].set_visible(False)
    gca().spines['top'].set_visible(False)

    subplot(4,1,2)
    plotit(r_,r_baseline,frac=0)
    #title('Offset between biomarker and biomass () ś 1.3')
    gca().set_xticklabels([])
    gca().set_xlabel('')
    gca().spines['left'].set_visible(False)
    gca().spines['top'].set_visible(False)
```

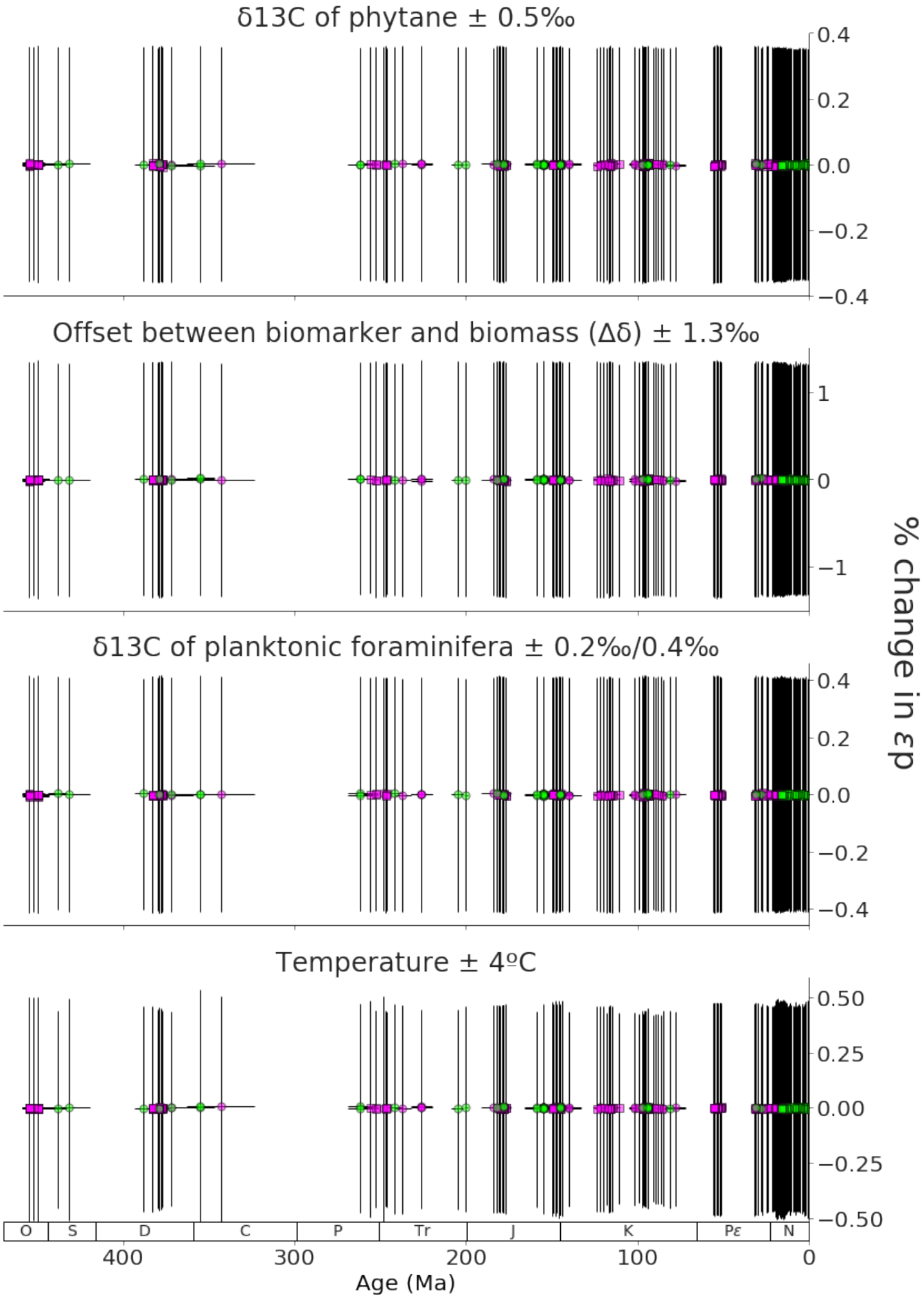
```
subplot(4,1,3)
plotit(r_a,r_baseline,frac=0)
#title('13C of planktonic foraminifera ś 0.2')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,4)
plotit(r_TC,r_baseline,frac=0.07)
#title('Temperature ś 4žC')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

gcf().text(0.99, 0.5, r'% change in $\epsilon$p', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig no titles percent change p uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


2
0
−2
10
0
−10
4
2
0
−2
−4
5.0
2.5
0.0
−2.5
−5.0
% change in εp
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)

4.0.3 same as above but absolute values rather than percent

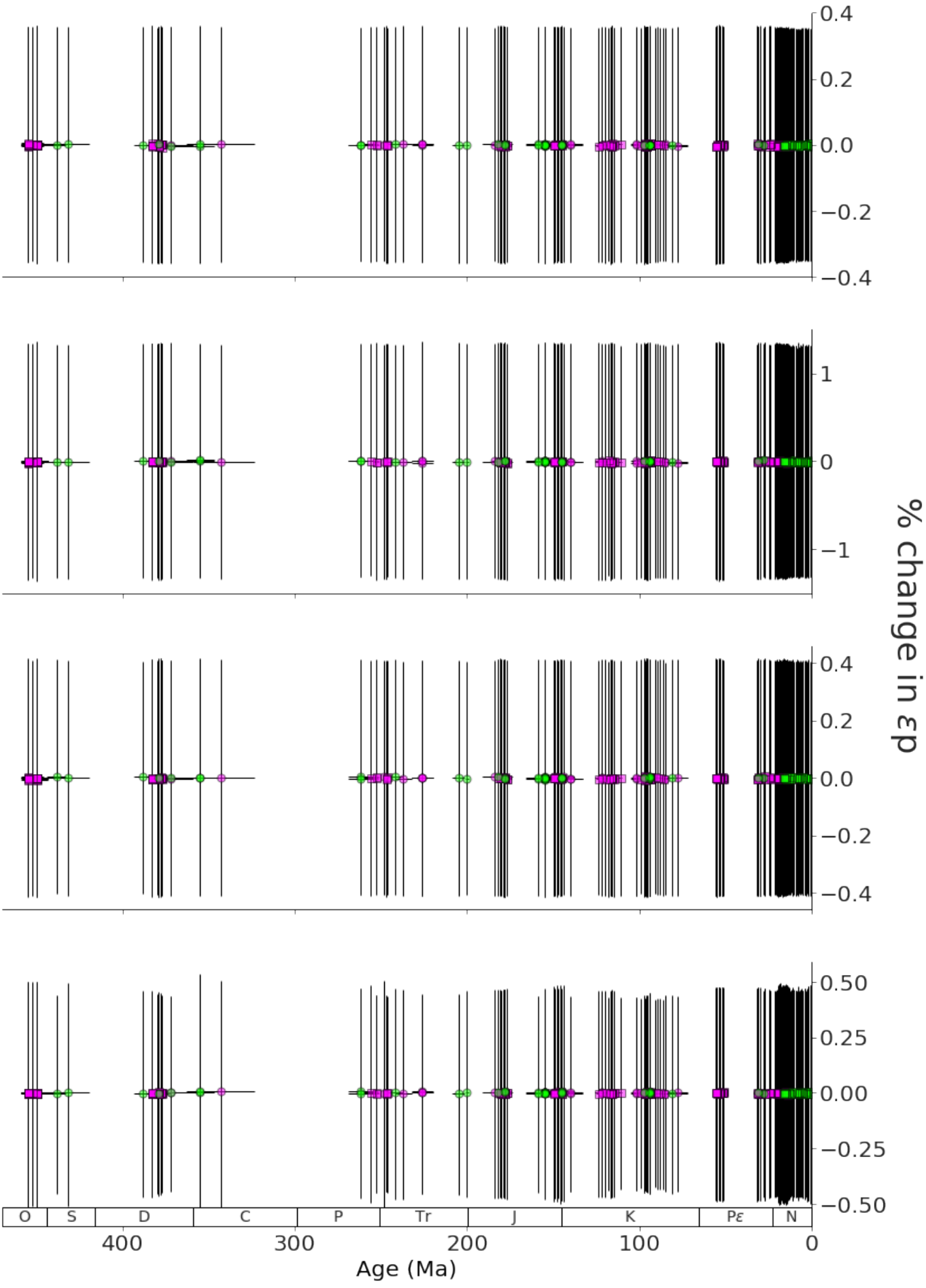
In [25]:
```
figure(figsize=(13,20))

        def plotit(r,r_baseline,frac=0.03):
            x=array(r['Ma'])
            xu=array(r['Ma_U'])
            xl=array(r['Ma_L'])

            idx_os_mo=r['index_other_oil']
            idx_os_ms=r['index_other_sedimentary']
            idx_cs_mo=r['index_current_oil']
            idx_cs_ms=r['index_current_sedimentary']


            val=(r['p']-r_baseline['p'])
            y=np.median(val,axis=1)
            yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
            yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


            xerr=array([x-xl,xu-x])
            yerr=array([y-yl,yu-y])

            alpha=0.5
            plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                label='Previously Published, Oil',alpha=alpha)
            plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                label='Previously Published, Sediment',alpha=alpha)
            plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                label='Current Study, Oil',alpha=alpha)
            plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                label='Current Study, Sediment',alpha=alpha)



            errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
            ax=gca()
            ax.invert_xaxis()
            xlim([470,0])
            xlabel('Age (Ma)')


            ax.yaxis.set_label_position("right")
            ax.yaxis.set_ticks_position("right")

            #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
            #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

            grid('off')

            add_time_periods(frac=frac)

            #==========

        subplot(4,1,1)
        plotit(r_Ph,r_baseline,frac=0)
        title('13C of phytane ś 0.5')
        gca().set_xticklabels([])
        gca().set_xlabel('')
        gca().spines['left'].set_visible(False)
        gca().spines['top'].set_visible(False)

        subplot(4,1,2)
        plotit(r_,r_baseline,frac=0)
        title('Offset between biomarker and biomass () ś 1.3')
        gca().set_xticklabels([])
        gca().set_xlabel('')
        gca().spines['left'].set_visible(False)
        gca().spines['top'].set_visible(False)
```

```
subplot(4,1,3)
plotit(r_a,r_baseline,frac=0)
title('13C of planktonic foraminifera ś 0.2/0.4')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,4)
plotit(r_TC,r_baseline,frac=0.07)
title('Temperature ś 4žC')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

gcf().text(0.99, 0.5, r'% change in $\epsilon$p', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig absolute change p uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


δ13C of phytane ± 0.5‰
0.4
0.2
0.0
−0.2
−0.4
Offset between biomarker and biomass (Δδ) ± 1.3‰
1
0
−1
δ13C of planktonic foraminifera ± 0.2‰/0.4‰
0.4
0.2
0.0
−0.2
−0.4
Temperature ± 4ºC
0.50
0.25
0.00
−0.25
−0.50
% change in εp
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)

In [26]:
```
figure(figsize=(13,20))

def plotit(r,r_baseline,frac=0.03):
    x=array(r['Ma'])
    xu=array(r['Ma_U'])
    xl=array(r['Ma_L'])

    idx_os_mo=r['index_other_oil']
    idx_os_ms=r['index_other_sedimentary']
    idx_cs_mo=r['index_current_oil']
    idx_cs_ms=r['index_current_sedimentary']


    val=(r['p']-r_baseline['p'])
    y=np.median(val,axis=1)
    yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
    yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


    xerr=array([x-xl,xu-x])
    yerr=array([y-yl,yu-y])

    alpha=0.5
    plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Oil',alpha=alpha)
    plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Sediment',alpha=alpha)
    plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Oil',alpha=alpha)
    plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Sediment',alpha=alpha)



    errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
    ax=gca()
    ax.invert_xaxis()
    xlim([470,0])
    xlabel('Age (Ma)')


    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")

    #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
    #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

    grid('off')

    add_time_periods(frac=frac)

    #==========

subplot(4,1,1)
plotit(r_Ph,r_baseline,frac=0)
#title('13C of phytane ś 0.5')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,2)
plotit(r_,r_baseline,frac=0)
#title('Offset between biomarker and biomass () ś 1.3')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,3)
plotit(r_a,r_baseline,frac=0)
#title('13C of planktonic foraminifera ś 0.2')
```

```
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(4,1,4)
plotit(r_TC,r_baseline,frac=0.07)
#title('Temperature ś ¿žC')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

gcf().text(0.99, 0.5, r'% change in $\epsilon$p', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig no titles absolute change p uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


0.4
0.2
0.0
−0.2
−0.4
1
0
−1
0.4
0.2
0.0
−0.2
−0.4
0.50
0.25
0.00
−0.25
−0.50
% change in εp
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)

4.1 actual value of ϵp for discrete values of parameters

In [27]:
```
# 13C of phytane ś 0.5   (Ph)
# Offset between biomarker and biomass () ś 1.3  (R)
# 13C of planktonic foraminifera ś 0.2/0.4  (a)
# Temperature ś 4žC  (TC)

Ph_arr=[-.5,-.25,.25,.5]
_arr=[3.50-1.28,3.50-1.28/2,3.50+1.28/2,3.50+1.28]
a_arr=[-.4,-.2,.2,.4]
TC_arr=[-4,-2,2,4]
```

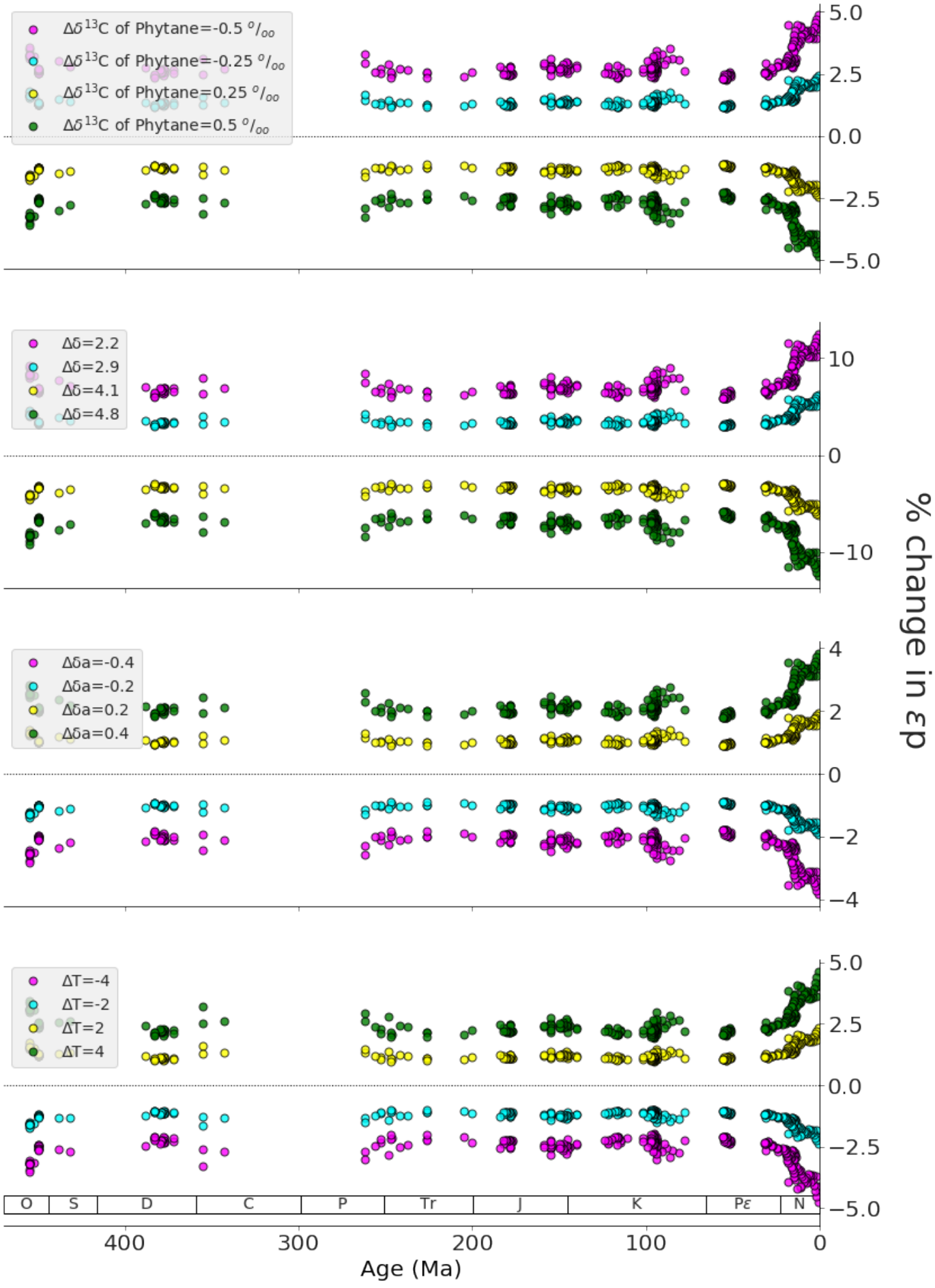
In [28]:
```
%%time
r_baseline=get_results(fix_parameters=['Ph','','a','TC'])

r_Ph=[]
for Ph in Ph_arr:
    r_Ph.append(get_results(fix_parameters=['Ph','','a','TC'],Ph=Ph))

r_=[]
for  in _arr:
    r_.append(get_results(fix_parameters=['Ph','','a','TC'],=))

r_a=[]
for a in a_arr:
    r_a.append(get_results(fix_parameters=['Ph','','a','TC'],a=a))


r_TC=[]
for TC in TC_arr:
    r_TC.append(get_results(fix_parameters=['Ph','','a','TC'],TC=TC))
```

```
Setting Ph to  -0.5
Setting Ph to  -0.25
Setting Ph to  0.25
Setting Ph to  0.5
Setting  to  2.2199999999999998
Setting  to  2.86
Setting  to  4.14
Setting  to  4.78
Setting a to  -0.4
Setting a to  -0.2
Setting a to  0.2
Setting a to  0.4
Setting TC to  -4
Setting TC to  -2
Setting TC to  2
Setting TC to  4
CPU times: user 4min, sys: 29 s, total: 4min 29s
Wall time: 4min 32s
```

In [29]:
```
figure(figsize=(13,20))

def plotit(r,r_baseline,frac=0.03,color=None):

    x=array(r['Ma'])
    xu=array(r['Ma_U'])
    xl=array(r['Ma_L'])

    idx_os_mo=r['index_other_oil']
    idx_os_ms=r['index_other_sedimentary']
    idx_cs_mo=r['index_current_oil']
    idx_cs_ms=r['index_current_sedimentary']


    val=(r['p']-r_baseline['p'])/r_baseline['p']*100
    y=np.median(val,axis=1)
```

```
    yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
    yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


    xerr=array([x-xl,xu-x])
    yerr=array([y-yl,yu-y])

    alpha=0.8

    if color is None:
        plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Oil',alpha=alpha)
        plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Sediment',alpha=alpha)
        plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Oil',alpha=alpha)
        plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Sediment',alpha=alpha)
    else:
        plot(x,y,'o',ms=7,color=color,markeredgecolor='k',markeredgewidth=1,alpha=alpha)


    #errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
    ax=gca()
    ax.invert_xaxis()
    xlim([470,0])
    xlabel('Age (Ma)')


    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")

    #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
    #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

    grid('off')

    add_time_periods(frac=frac)

    #==========

subplot(4,1,1)

for r,c in zip(r_Ph,['magenta','cyan','yellow','green']):
    plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

LL=legend([r'$\delta^{13}$C of Phytane=%g $^{o}/_{oo}$' % Ph for Ph in Ph_arr],
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')

gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


subplot(4,1,2)

for r,c in zip(r_,['magenta','cyan','yellow','green']):
    plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['=%.1f' %  for  in _arr],
```

```
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')

subplot(4,1,3)

for r,c in zip(r_a,['magenta','cyan','yellow','green']):
    plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['a=%.1f' % a for a in a_arr],
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')

subplot(4,1,4)

for r,c in zip(r_TC,['magenta','cyan','yellow','green']):
    if c=='green':
        plotit(r,r_baseline,frac=0.07,color=c)
    else:
        plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['T=%d' % T for T in TC_arr],
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')


gcf().text(0.99, 0.5, r'% change in $\epsilon$p', fontsize=30,
           ha='center', va='center', rotation=270)

if saveit:
    plt.savefig("../figures/Fig percent change epsilonp uncertainty associated with discrete values of each
parameter.png",dpi=900,bbox_inches='tight')
```


Δδ13C of Phytane=-0.5 o/oo
Δδ13C of Phytane=-0.25 o/oo
Δδ13C of Phytane=0.25 o/oo
Δδ13C of Phytane=0.5 o/oo
5.0
2.5
0.0
−2.5
−5.0
Δδ=2.2
Δδ=2.9
Δδ=4.1
Δδ=4.8
10
0
−10
Δδa=-0.4
Δδa=-0.2
Δδa=0.2
Δδa=0.4
4
2
0
−2
−4
ΔT=-4
ΔT=-2
ΔT=2
ΔT=4
5.0
2.5
0.0
−2.5
−5.0
% change in εp
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)

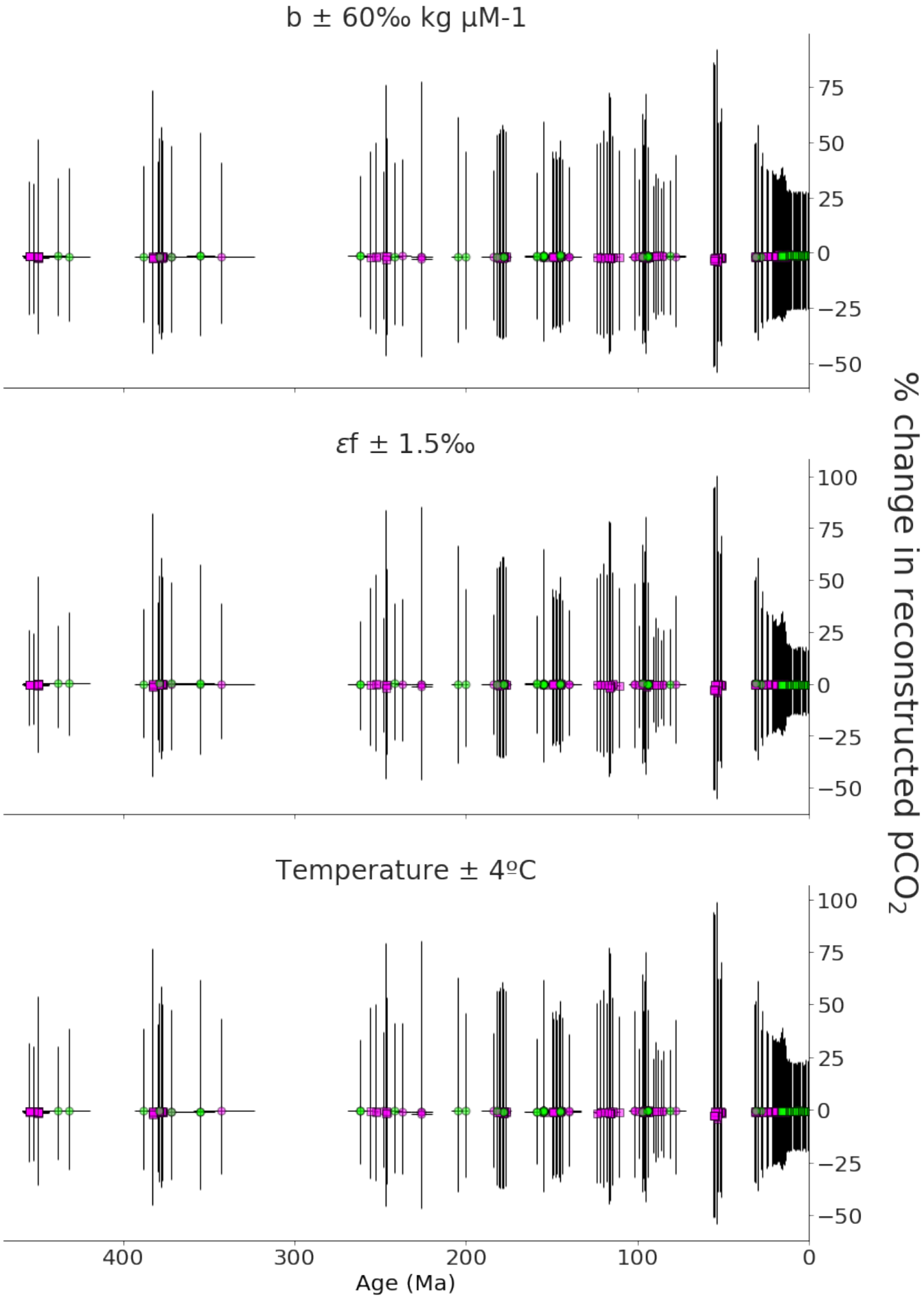
5 *Figure pCO2[aq] from phytane breakdown on the uncertainty associated with each individual parameter*

5.0.1 , expressed in a 2-part figure based on the three parameters making up the p equation

```
CO2[aq] = b / (f - p)

pCO2 = [CO2(aq)] / K0

A: b ś 60 kg ţM-1 (ranging from about 120-230)
B: f ś 1.5 (ranging from 25-28)
C: K0: Temperature ś 4žC

A. b (influence on ppm) B. b (influence on %)
C. f (ppm) D. f (%)
E. K0 temperature (ppm) F. K0 temperature (%)
```

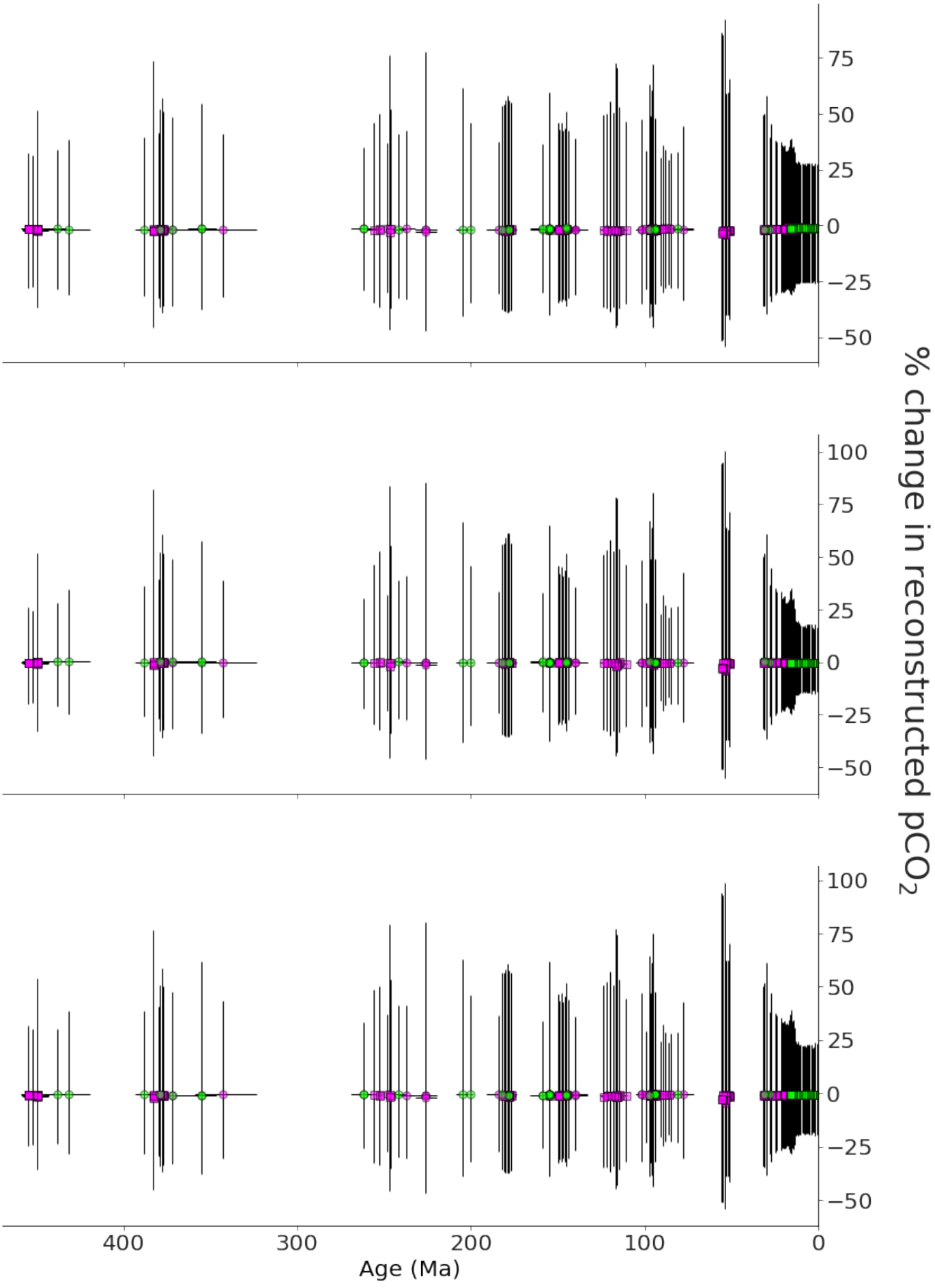
```
In [30]: %%time
         r_baseline=get_results(fix_parameters=['b','f','TC'])
         r_b=get_results(fix_parameters=['f','TC'])
         r_f=get_results(fix_parameters=['b','TC'])
         r_TC=get_results(fix_parameters=['b','f'])

CPU times: user 1min 3s, sys: 10 s, total: 1min 13s
Wall time: 1min 15s


In [31]: figure(figsize=(13,20))

         def plotit(r,r_baseline,frac=0.03):
             x=array(r['Ma'])
             xu=array(r['Ma_U'])
             xl=array(r['Ma_L'])

             idx_os_mo=r['index_other_oil']
             idx_os_ms=r['index_other_sedimentary']
             idx_cs_mo=r['index_current_oil']
             idx_cs_ms=r['index_current_sedimentary']


             val=(r['pCO2']-r_baseline['pCO2'])/r_baseline['pCO2']*100
             y=np.median(val,axis=1)
             yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
             yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


             xerr=array([x-xl,xu-x])
             yerr=array([y-yl,yu-y])

             alpha=0.5
             plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                 label='Previously Published, Oil',alpha=alpha)
             plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                 label='Previously Published, Sediment',alpha=alpha)
             plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                 label='Current Study, Oil',alpha=alpha)
             plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                 label='Current Study, Sediment',alpha=alpha)



             errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
             ax=gca()
```

```
    ax.invert_xaxis()
    xlim([470,0])
    xlabel('Age (Ma)')


    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")

    #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
    #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

    grid('off')

    add_time_periods(frac=frac)

    #==========

subplot(3,1,1)
plotit(r_b,r_baseline,frac=0)
title('b ś 60 kg ţM-1')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(3,1,2)
plotit(r_f,r_baseline,frac=0)
title(r'$\epsilon$f ś 1.5')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(3,1,3)
plotit(r_TC,r_baseline,frac=0)
title('Temperature ś 4žC')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


gcf().text(0.99, 0.5, r'% change in reconstructed pCO$_2$', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig percent change pCO2 uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


b ± 60‰ kg µM-1
εf ± 1.5‰
Temperature ± 4ºC
% change in reconstructed pCO2
75
50
25
0
−25
−50
100
400
300
200
100
0
Age (Ma)

In [32]:
```
figure(figsize=(13,20))

def plotit(r,r_baseline,frac=0.03):
    x=array(r['Ma'])
    xu=array(r['Ma_U'])
    xl=array(r['Ma_L'])

    idx_os_mo=r['index_other_oil']
    idx_os_ms=r['index_other_sedimentary']
    idx_cs_mo=r['index_current_oil']
    idx_cs_ms=r['index_current_sedimentary']


    val=(r['pCO2']-r_baseline['pCO2'])/r_baseline['pCO2']*100
    y=np.median(val,axis=1)
    yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
    yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


    xerr=array([x-xl,xu-x])
    yerr=array([y-yl,yu-y])

    alpha=0.5
    plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Oil',alpha=alpha)
    plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Sediment',alpha=alpha)
    plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Oil',alpha=alpha)
    plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Sediment',alpha=alpha)



    errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
    ax=gca()
    ax.invert_xaxis()
    xlim([470,0])
    xlabel('Age (Ma)')


    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")

    #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
    #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

    grid('off')

    add_time_periods(frac=frac)

    #==========

subplot(3,1,1)
plotit(r_b,r_baseline,frac=0)
#title('b ś 60 kg ţM-1')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(3,1,2)
plotit(r_f,r_baseline,frac=0)
#title(r'$\epsilon$f ś 1.5')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(3,1,3)
plotit(r_TC,r_baseline,frac=0)
#title('Temperature ś ţžC')
```

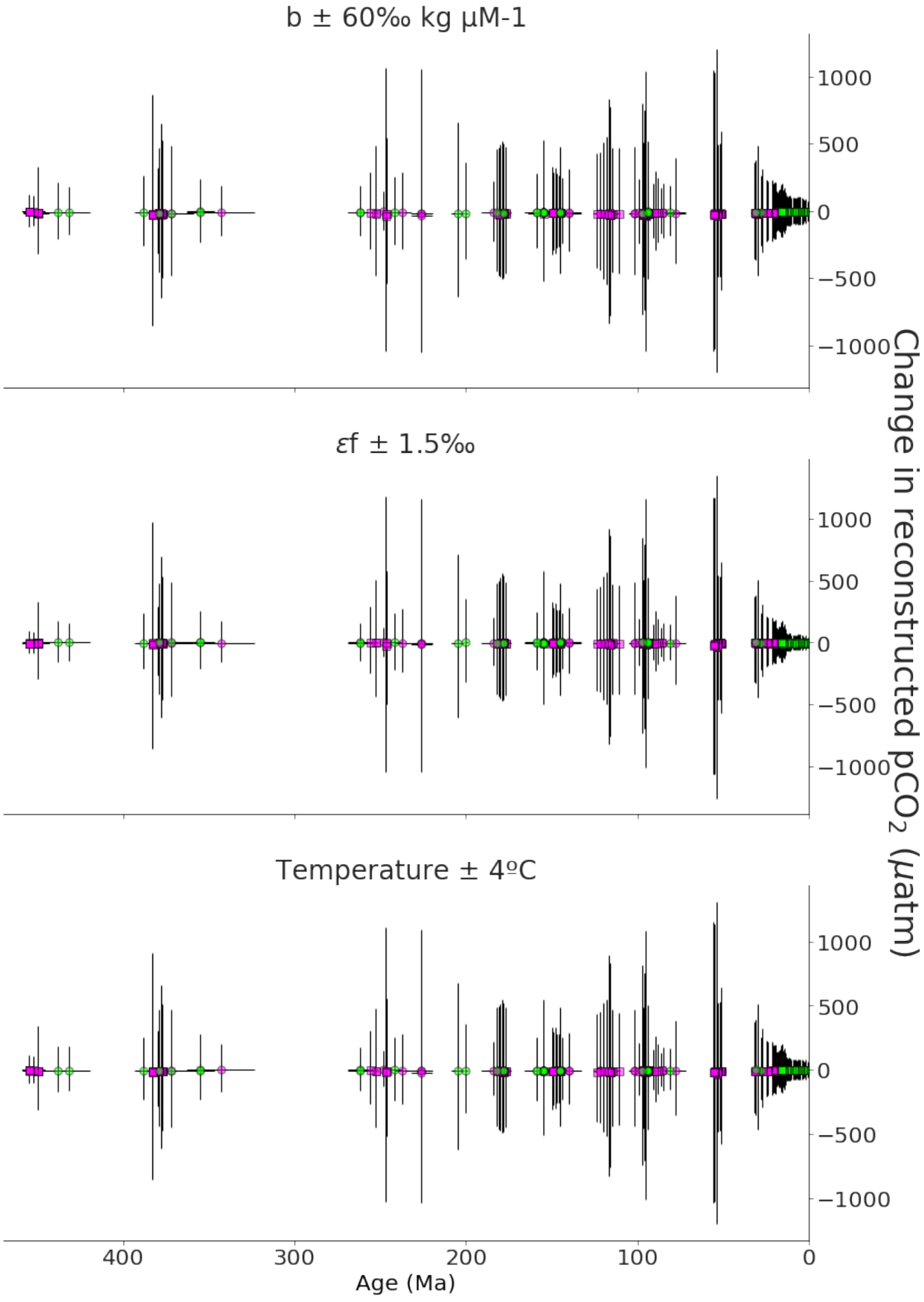
```
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


gcf().text(0.99, 0.5, r'% change in reconstructed pCO$_2$', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig no titles percent change pCO2 uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


75
50
25
0
−25
−50
100
75
50
25
0
−25
−50
100
75
50
25
0
−25
−50
400
300
200
100
0
Age (Ma)
% change in reconstructed pCO_2

5.0.2 Absolute changes in pCO_2

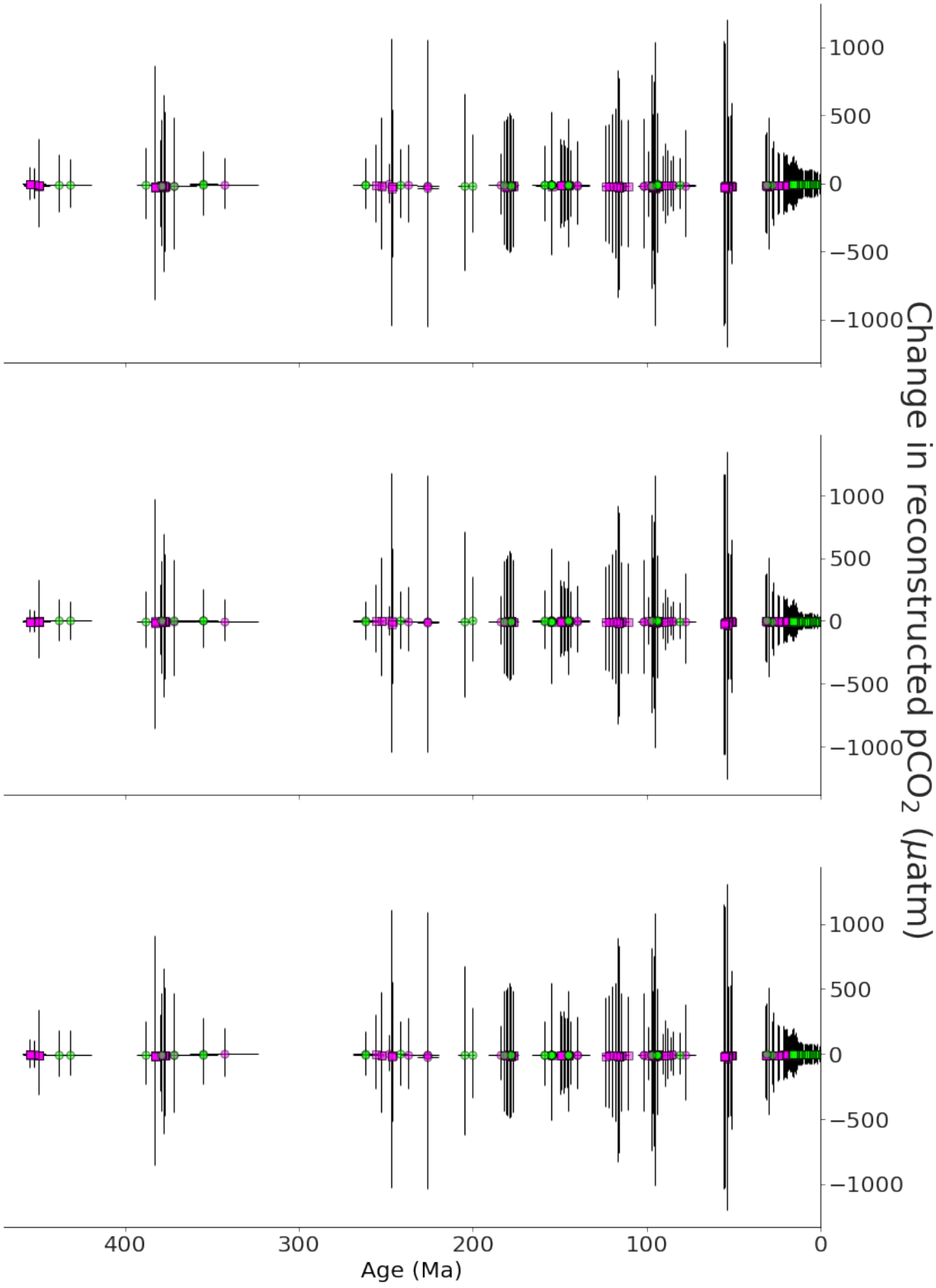
In [33]:
```
figure(figsize=(13,20))

    def plotit(r,r_baseline,frac=0.03):
        x=array(r['Ma'])
        xu=array(r['Ma_U'])
        xl=array(r['Ma_L'])

        idx_os_mo=r['index_other_oil']
        idx_os_ms=r['index_other_sedimentary']
        idx_cs_mo=r['index_current_oil']
        idx_cs_ms=r['index_current_sedimentary']


        val=(r['pCO2']-r_baseline['pCO2'])
        y=np.median(val,axis=1)
        yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
        yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


        xerr=array([x-xl,xu-x])
        yerr=array([y-yl,yu-y])

        alpha=0.5
        plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Oil',alpha=alpha)
        plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Sediment',alpha=alpha)
        plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Oil',alpha=alpha)
        plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Sediment',alpha=alpha)



        errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
        ax=gca()
        ax.invert_xaxis()
        xlim([470,0])
        xlabel('Age (Ma)')


        ax.yaxis.set_label_position("right")
        ax.yaxis.set_ticks_position("right")

        #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
        #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

        grid('off')

        add_time_periods(frac=frac)

        #==========

    subplot(3,1,1)
    plotit(r_b,r_baseline,frac=0)
    title('b ś 60 kg ţM-1')
    gca().set_xticklabels([])
    gca().set_xlabel('')
    gca().spines['left'].set_visible(False)
    gca().spines['top'].set_visible(False)

    subplot(3,1,2)
    plotit(r_f,r_baseline,frac=0)
    title(r'$\epsilon$f ś 1.5')
    gca().set_xticklabels([])
    gca().set_xlabel('')
    gca().spines['left'].set_visible(False)
    gca().spines['top'].set_visible(False)
```

```
subplot(3,1,3)
plotit(r_TC,r_baseline,frac=0)
title('Temperature ś 4žC')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


gcf().text(0.99, 0.5, r'Change in reconstructed pCO$_2$ ($\mu$atm)', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig absolute change pCO2 uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


b ± 60‰ kg µM-1
εf ± 1.5‰
Temperature ± 4ºC
1000
500
0
−500
−1000
400
300
200
100
0
Age (Ma)
Change in reconstructed pCO2 (µatm)

In [34]:
```
figure(figsize=(13,20))

def plotit(r,r_baseline,frac=0.03):
    x=array(r['Ma'])
    xu=array(r['Ma_U'])
    xl=array(r['Ma_L'])

    idx_os_mo=r['index_other_oil']
    idx_os_ms=r['index_other_sedimentary']
    idx_cs_mo=r['index_current_oil']
    idx_cs_ms=r['index_current_sedimentary']


    val=(r['pCO2']-r_baseline['pCO2'])
    y=np.median(val,axis=1)
    yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
    yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


    xerr=array([x-xl,xu-x])
    yerr=array([y-yl,yu-y])

    alpha=0.5
    plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Oil',alpha=alpha)
    plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
        label='Previously Published, Sediment',alpha=alpha)
    plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Oil',alpha=alpha)
    plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
        label='Current Study, Sediment',alpha=alpha)



    errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
    ax=gca()
    ax.invert_xaxis()
    xlim([470,0])
    xlabel('Age (Ma)')


    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")

    #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
    #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

    grid('off')

    add_time_periods(frac=frac)

    #==========

subplot(3,1,1)
plotit(r_b,r_baseline,frac=0)
#title('b ś 60 kg ţM-1')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(3,1,2)
plotit(r_f,r_baseline,frac=0)
#title(r'$\epsilon$f ś 1.5')
gca().set_xticklabels([])
gca().set_xlabel('')
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

subplot(3,1,3)
plotit(r_TC,r_baseline,frac=0)
#title('Temperature ś ţžC')
```

```
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


gcf().text(0.99, 0.5, r'Change in reconstructed pCO$_2$ ($\mu$atm)', fontsize=30,
           ha='center', va='center', rotation=270)


if saveit:
    plt.savefig("../figures/Fig no titles absolute change pCO2 uncertainty associated with each
parameter.png",dpi=900,bbox_inches='tight')
```


1000
500
0
−500
−1000
Change in reconstructed pCO_2 (μatm)
400
300
200
100
0
Age (Ma)

5.1 Discrete changes in parameters

In [35]:
```
b_arr=[170-70,170-35,170+35,170+70]
f_arr=[26.5-1.5,26.5-0.75,26.5+0.75,26.5+1.5]
TC_arr=[-4,-2,2,4]
```

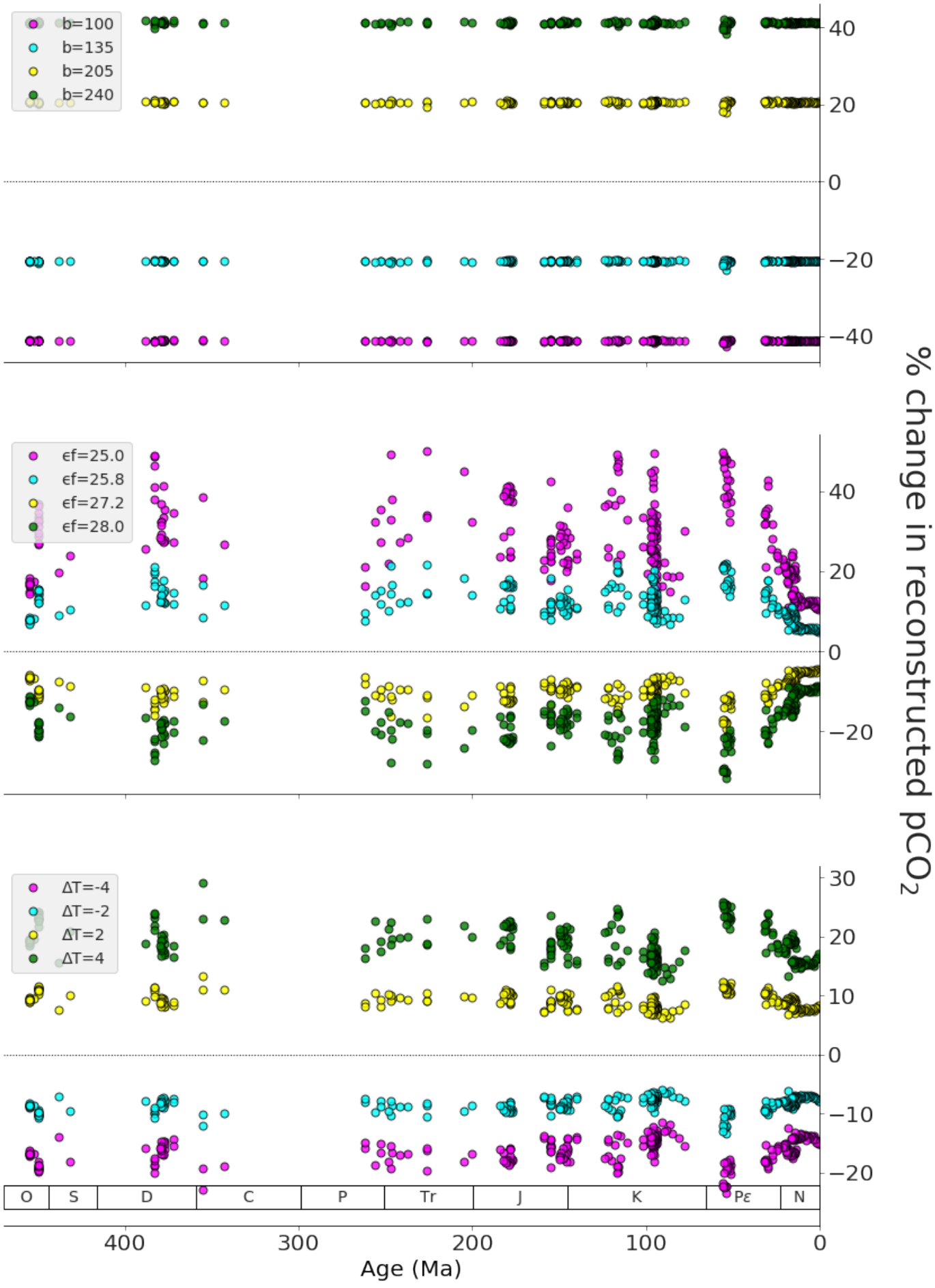
In [36]:
```
%%time
r_baseline=get_results(fix_parameters=['b','f','TC'])

r_b=[]
for b in b_arr:
    r_b.append(get_results(fix_parameters=['f','TC'],b=b))

r_f=[]
for f in f_arr:
    r_f.append(get_results(fix_parameters=['b','TC'],f=f))

r_TC=[]
for TC in TC_arr:
    r_TC.append(get_results(fix_parameters=['f','b','TC'],TC=TC))
```

```
Setting b to  100
Setting b to  135
Setting b to  205
Setting b to  240
Setting f to  25.0
Setting f to  25.75
Setting f to  27.25
Setting f to  28.0
Setting TC to  -4
Setting TC to  -2
Setting TC to  2
Setting TC to  4
CPU times: user 2min 59s, sys: 19 s, total: 3min 18s
Wall time: 3min 22s
```

In [37]:
```
figure(figsize=(13,20))

def plotit(r,r_baseline,frac=0.03,color=None):

    x=array(r['Ma'])
    xu=array(r['Ma_U'])
    xl=array(r['Ma_L'])

    idx_os_mo=r['index_other_oil']
    idx_os_ms=r['index_other_sedimentary']
    idx_cs_mo=r['index_current_oil']
    idx_cs_ms=r['index_current_sedimentary']


    val=(r['pCO2']-r_baseline['pCO2'])/r_baseline['pCO2']*100
    y=np.median(val,axis=1)
    yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
    yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


    xerr=array([x-xl,xu-x])
    yerr=array([y-yl,yu-y])

    alpha=0.8

    if color is None:
        plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Oil',alpha=alpha)
        plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
            label='Previously Published, Sediment',alpha=alpha)
        plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
```

```
            label='Current Study, Oil',alpha=alpha)
        plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
            label='Current Study, Sediment',alpha=alpha)
    else:
        plot(x,y,'o',ms=7,color=color,markeredgecolor='k',markeredgewidth=1,alpha=alpha)


    #errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
    ax=gca()
    ax.invert_xaxis()
    xlim([470,0])
    xlabel('Age (Ma)')


    ax.yaxis.set_label_position("right")
    ax.yaxis.set_ticks_position("right")

    #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
    #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%}$)',rotation=270,va='bottom', labelpad=20)

    grid('off')

    add_time_periods(frac=frac)

    #==========

subplot(3,1,1)

for r,c in zip(r_b,['magenta','cyan','yellow','green']):
    plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

LL=legend(['b=%d' % b for b in b_arr],
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')

gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


subplot(3,1,2)

for r,c in zip(r_f,['magenta','cyan','yellow','green']):
    plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['f=%.1f' % f for f in f_arr],
          loc='best',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')


subplot(3,1,3)

for r,c in zip(r_TC,['magenta','cyan','yellow','green']):
    if c=='green':
        plotit(r,r_baseline,frac=0.07,color=c)
    else:
        plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)
```

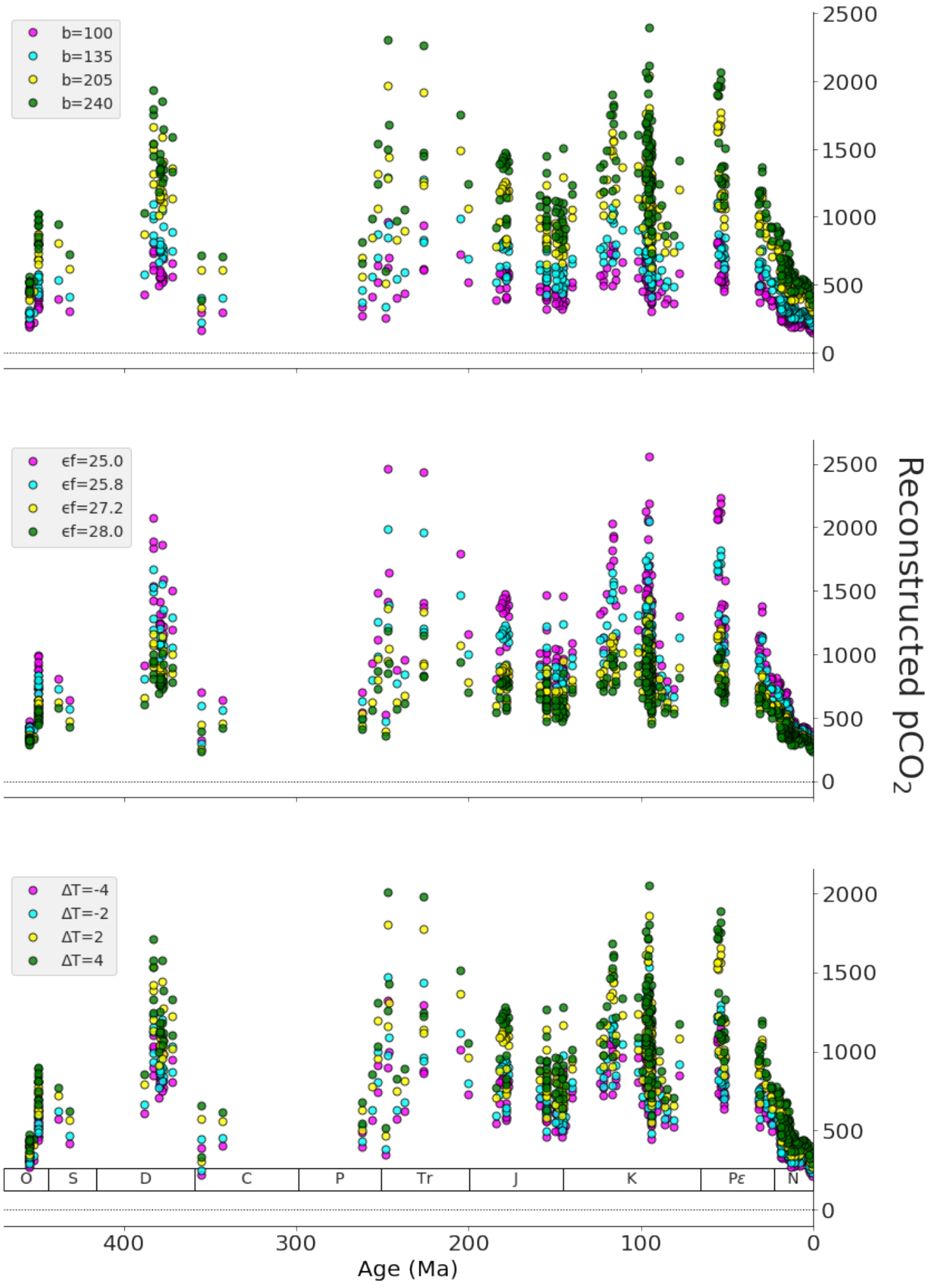
```
gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['T=%d' % T for T in TC_arr],
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')


gcf().text(0.99, 0.5, r'% change in reconstructed pCO$_2$', fontsize=30,
           ha='center', va='center', rotation=270)

if saveit:
    plt.savefig("../figures/Fig percent change pCO2 uncertainty associated with discrete values of each
parameter.png",dpi=900,bbox_inches='tight')
```


b=100
b=135
b=205
b=240
ϵf=25.0
ϵf=25.8
ϵf=27.2
ϵf=28.0
ΔT=-4
ΔT=-2
ΔT=2
ΔT=4
40
20
0
−20
−40
30
10
−10
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)
% change in reconstructed pCO_2

5.2 absolute values

In [38]:
```
figure(figsize=(13,20))

    def plotit(r,r_baseline,frac=0.03,color=None):

        x=array(r['Ma'])
        xu=array(r['Ma_U'])
        xl=array(r['Ma_L'])

        idx_os_mo=r['index_other_oil']
        idx_os_ms=r['index_other_sedimentary']
        idx_cs_mo=r['index_current_oil']
        idx_cs_ms=r['index_current_sedimentary']


        val=r['pCO2']
        y=np.median(val,axis=1)
        yl=np.percentile(val,(100-errorbar_level)/2,axis=1)
        yu=np.percentile(val,100-(100-errorbar_level)/2,axis=1)


        xerr=array([x-xl,xu-x])
        yerr=array([y-yl,yu-y])

        alpha=0.8

        if color is None:
            plot(x[idx_os_mo],y[idx_os_mo],'o',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                label='Previously Published, Oil',alpha=alpha)
            plot(x[idx_os_ms],y[idx_os_ms],'s',ms=7,color=color1,markeredgecolor='k',markeredgewidth=1,
                label='Previously Published, Sediment',alpha=alpha)
            plot(x[idx_cs_mo],y[idx_cs_mo],'o',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                label='Current Study, Oil',alpha=alpha)
            plot(x[idx_cs_ms],y[idx_cs_ms],'s',ms=7,color=color2,markeredgecolor='k',markeredgewidth=1,
                label='Current Study, Sediment',alpha=alpha)
        else:
            plot(x,y,'o',ms=7,color=color,markeredgecolor='k',markeredgewidth=1,alpha=alpha)


        #errorbar(x,y,xerr=xerr,yerr=yerr,fmt='none',lw=1,zorder=0,ecolor='k',capsize=0)
        ax=gca()
        ax.invert_xaxis()
        xlim([470,0])
        xlabel('Age (Ma)')


        ax.yaxis.set_label_position("right")
        ax.yaxis.set_ticks_position("right")

        #h=ylabel(r'% change in p ($^{o}/_{oo}$)',rotation=270,va='bottom', labelpad=20)
        #h=ylabel(r'$\delta^{13}$C of Phytane ($^{\%}/_{\%\%}$)',rotation=270,va='bottom', labelpad=20)

        grid('off')

        add_time_periods(frac=frac)

        #==========

    subplot(3,1,1)

    for r,c in zip(r_b,['magenta','cyan','yellow','green']):
        plotit(r,r_baseline,frac=0,color=c)

    xl=gca().set_xlim()
    plot(xl,[0,0],'k:',linewidth=1)

    LL=legend(['b=%d' % b for b in b_arr],
              loc='upper left',frameon=True,fontsize=14)
    LL.get_frame().set_facecolor('#EEEEEE')
```

```
gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)


subplot(3,1,2)

for r,c in zip(r_f,['magenta','cyan','yellow','green']):
    plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

gca().set_xticklabels([])
gca().set_xlabel('')

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['f=%.1f' % f for f in f_arr],
          loc='best',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')


subplot(3,1,3)

for r,c in zip(r_TC,['magenta','cyan','yellow','green']):
    if c=='green':
        plotit(r,r_baseline,frac=0.07,color=c)
    else:
        plotit(r,r_baseline,frac=0,color=c)

xl=gca().set_xlim()
plot(xl,[0,0],'k:',linewidth=1)

gca().spines['left'].set_visible(False)
gca().spines['top'].set_visible(False)

LL=legend(['T=%d' % T for T in TC_arr],
          loc='upper left',frameon=True,fontsize=14)
LL.get_frame().set_facecolor('#EEEEEE')


gcf().text(0.99, 0.5, r'Reconstructed pCO$_2$', fontsize=30,
           ha='center', va='center', rotation=270)

if saveit:
    plt.savefig("../figures/Fig pCO2 uncertainty associated with discrete values of each
parameter.png",dpi=900,bbox_inches='tight')
```


b=100
b=135
b=205
b=240
ϵf=25.0
ϵf=25.8
ϵf=27.2
ϵf=28.0
ΔT=-4
ΔT=-2
ΔT=2
ΔT=4
2500
2000
1500
1000
500
0
O
S
D
C
P
Tr
J
K
Pε
N
400
300
200
100
0
Age (Ma)
Reconstructed pCO2