

## Development of an algorithm to infer the free hydrocarbon content (S1) from well logs

Edson Alonso Falla Luza<sup>1</sup>, Maria Luiza Cyrino Paiva<sup>1</sup>, Hélio Sérgio Rocha Lima e Silva<sup>2</sup>, Antonio Fernando Menezes Freire<sup>1</sup>.

<sup>1</sup> Universidade Federal Fluminense – UFF

<sup>2</sup> Independent Consultant

Copyright 2019, SBGf - Sociedade Brasileira de Geofísica

This paper was prepared for presentation during the 16<sup>th</sup> International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 19-22 August 2019.

Contents of this paper were reviewed by the Technical Committee of the 16<sup>th</sup> International Congress of the Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of the Brazilian Geophysical Society is prohibited.

### Abstract

The evaluation of the exploratory potential for shale oil/shale gas of a source rock involves the characterization of three key factors: existence of high TOC, adequate maturity and susceptibility to fracking. Additional factors such as depositional and geological control over the distribution of organic matter in the source rock should also be investigated. The free hydrocarbon content (S1) is one of the Rock Eval pyrolysis data and it is measured in laboratory in rock samples collected in wells (cuttings, lateral samples or cores) or outcrops. The problem is that not all wells in a given region have sampling for geochemical analysis, restricting this information to a few wells. However, most of the wells have the GR and LLD logs, which led the authors to develop an algorithm to infer the parameter S1 synthetic (S1s), similar to what was done by Passey and co-authors in the development of the  $\Delta\log R$  calculation for TOC. The calculation of the S1s, calculated from the GR and LLD logs, constitute a considerable tool for the exploratory assessment of the potential for oil production in unconventional reservoirs, or to evaluate source rocks.

### Introduction

The study area is located in the northern portion of the Campos Basin, passive margin basin formed from a rift. It is located between the Vitória and Cabo Frio highs (Fig. 1) with an area of approximately 100,000 km<sup>2</sup>. The Baleia Azul Field belonging to the Parque das Baleias Cluster, a set of oil and gas fields located at the northern part of the basin (Fig. 2).

The sedimentary succession of this basin occurred in 3 main phases (Winter et al., 2007). The first is the rift phase with predominantly continental sedimentation. The second is the transitional phase, or sag phase, when the evaporites were deposited. The third is the drift phase with 3 distinct deposition regimes: carbonate platforms, transgressive marine and regressive marine.

The Baleia Azul oil field belongs to the exploratory block BC-60 entitled Parque das Baleias. It is located approximately 80km from the coast of Espírito Santo state with an area of 63.69km<sup>2</sup>, where the water depth varies between 1200m in the proximal area and 1500m in the distal area.

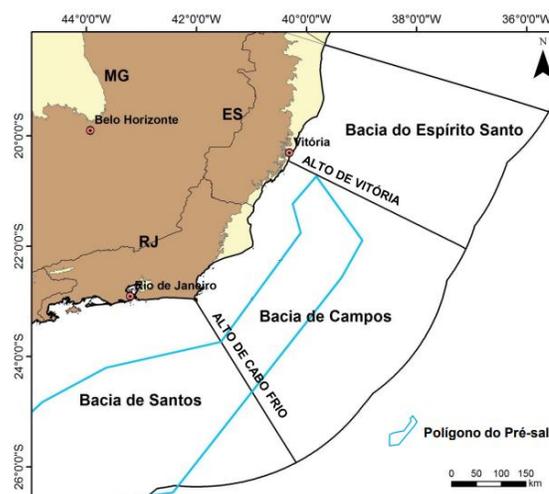


Figure 1: Location map of Campos Basin, SE Brazil. Modified from ANP (Pereira, 2015).

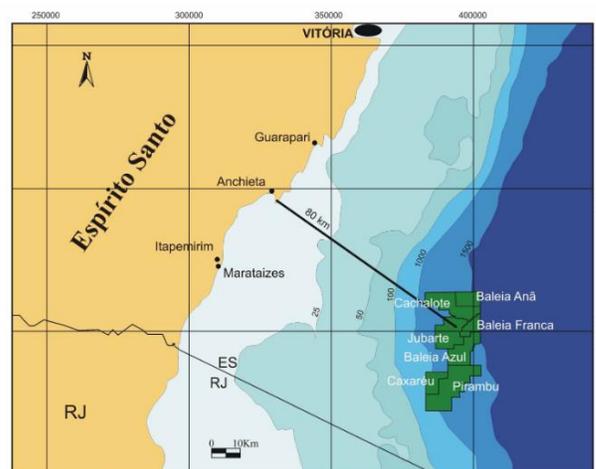


Figure 2: Location map of set of Oil/Gas Fields Parque das Baleias in the Campos Basin, SE Brazil. Modified from ANP/BDEP WebMaps (2019).

The main pre-salt source rocks in the Campos Basin are the Buracica-Jiquiá shales of the Barremian-Aptian of the Lagoa Feia group deposited in the rift stage in a lacustrine environment. Such shales have TOCs of 2-6%, kerogen type I and produce oil ranging from 17-37° API. It is not ruled out the possibility of source rocks among the marine sediments of the post salt.

## Free hydrocarbon content (S1) from well logs

The main reservoirs of the Baleia Azul field are the microbial carbonates of the Macabu Formation of Alagoas (Aptian) and turbidite sandstones of the Carapebus Formation (Santonian-Miocene) with maximum porosity values varying from 20-32% and permeability between 1D -5D.

The seal rocks are shales of the Coqueiros Fm. (Aptian), evaporites of Retiro Fm. (Aptian), limestones and shales of the Outeiro Fm. (Albian-Cenomanian), and shales of the Ubatuba Fm. (Turonian-recent). The inferred migration routes are: salt windows (where there are no evaporites from the Retiro Fm.), direct contact or lateral contact due to faults.

Throughout the evolution of the basin, structural, stratigraphic and mixed traps have been identified: quaquaversal structures, rotated failed blocks, rollover faults, sandy canalized reservoirs.

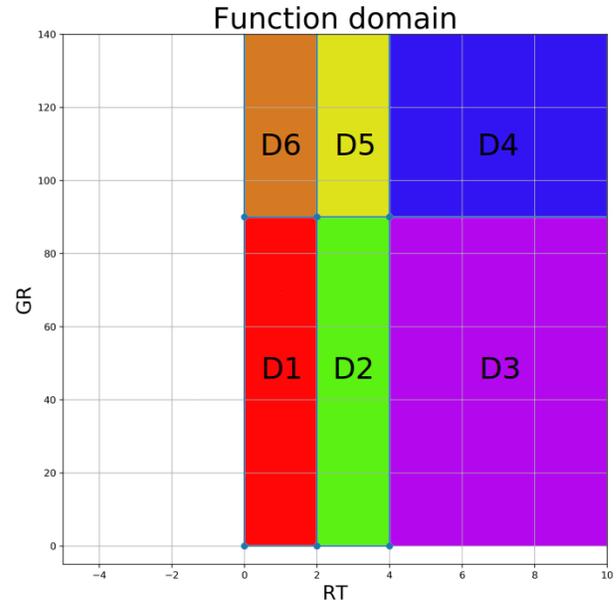
### Method

This work proposes a Python implementation (Appendix A) and a workflow for the application and generalization of an algorithm to calculate a S1 synthetic (S1s). This algorithm was initially proposed by Hélio Sérgio Rocha Lima e Silva (personal communication) and applied in shales of Candeias Fm. in the Recôncavo Basin, NE Brazil. The algorithm is based on the assumption that exists a direct relation between the free hydrocarbon content (S1), given by pyrolysis analysis, and Gamma Ray (GR) values, as well as an inverse relation between S1 and resistivity (LLD or ILD) values.

The algorithm is, indeed, divide in 3 algorithms. Each one is responsible for the calculation of a S1s for different levels of correlation (high, medium, low). For now, we are going to focus our attention on the high correlation curve (S1\_P90). Everything sad down here for the S1\_P90 curve applies for the medium (S1\_P50) and low (S1\_P10) curves as well. The S1\_P90 algorithm is based on the RT and GR limits, in order to eliminate the effect of the mineral matrix variation for the resistivity response. That being sad, the algorithm is subdivided in many different scenarios. In mathematical terms, the function responsible for the S1\_P90 calculation is a function defined by parts, whit no discontinuity. That means that the domain of the function is not a full set but, in this case, 6 subsets of the plane (Fig. 3), each one defining only a function.

Our implementation takes advantage of this observation. Usually, a defined by parts function need a lot of "if" functions to be implemented. In our case, since the domains are disjointed, we will create a new function for each domain, and defined each function as 0 outside the correspondent domain. For example, the algorithm is divided in 6 domains, called here D1, D2, ..., D6 (Fig. 03). For D1, we are going to create a function f1 which domain is all the cartesian plane, but it only has a non-zero value in D1. After all, our final function will be  $f(x,y)$ , defined in all the cartesian plane, but only having value in D1, D2, ..., D6.

$$f(x, y) = \sum_{i=1}^6 f_i(x, y),$$



**Figure 3:** Example of how a function defined by parts domain works.

The workflow is composed of 4 phases (Fig. 4):

**Phase 1** - In this phase, we select the corresponding values to the depth of the analytical S1 (from cuttings) in the GR and LLD curve. It has been developed and optimized an algorithm in Python in order to perform this task quickly and automatically. Since well log data are usually dense, we implement the algorithm in a logic way, so it wasn't inefficient in the sense of processing machine time.

**Phase 2** - Once the S1s algorithm propose a relation and not an equality, we are going to try to find the proportionality coefficients by doing the required adjustments when implementing the code in Python. Since the S1s algorithm was firstly developed to study the Candeias Fm., there was a need to perform some adjustments in the final S1s curve. In this work, we suggest two adjustments, independents one from each other in the sense that not necessarily both must be used, that show sufficient to generate a satisfactory S1s curve.

The first one is just a divisibility term, and this must be done first with the discrete points, in order to adjust the S1s values generated by this algorithm with the analytical values of S1 itself. The second one is based on the empirical fact observed that the more the GR curve oscillates the more S1s curve also will oscillate. We utilized, in the case of high frequency GR curve, a smoothing IIR filter with mirror-symmetric boundary conditions using a cascade of first-order sections. The mathematical formalism is not necessary. For now, the filter is just a way to smooth a signal without losing information.

This works for any signal and its purely based on the frequency spectral content of the signal itself.

All this process was made by the implementation of a Python code. This is a transitive and recursively phase, and it must not be a surprise that it must be done some times. Indeed, we recommend this phase to be done again after phase 3.

**Phase 3** - Since the filter mentioned before is not a linear function, there is a difference between smooth the signals and operate them and operate the signals and then smooth the result. This phase is related to the first option.

In this phase we applied the IIR filter on both the GR and the LLD curves so we can ignore the high oscillatory behavior of both and focus just in its numerical information. Once we applied the filter, we then utilize the S1s algorithm and save the result.

**Phase 4** - This phase is related to the second option from the two mentioned on phase 3.

The IIR filter is not applied in the GR and LLD curves, but in the S1s curve coming from the S1s algorithm. It is worth to remember that this is done so the best result can be chosen, once there exist a big difference between phase 3 and phase 4.

Once the S1s algorithm differentiates the P\_90, P\_50 and P\_10 curves, we must apply it to all the 3 curves. That being said, we must apply the flowchart for the S1\_P90, S1\_P50, and S1\_P10 curves, compare, and chose the one that more suites to the S1 analytical measurements.

The well 4-BRSA-420-ESS, located in the Baleia Azul Field, was chosen to test the method for having a reasonable amount of Rock-Eval pyrolysis data. During the use of our workflow in the 4-BRSA-420-ESS well, the phase 1 occurred in the expected way. The algorithm developed to do the discretization of both GR and LLD curves perform its job in about 0.01 seconds, which is clearly faster than doing it by hand. Phase 2 was surprisingly satisfactory. Lima e Silva (personal communication) was not sure if the S1s algorithm would work outside the lacustrine shales of the Candeias Fm. The first result of this work is to confirm the functionality of S1s algorithm in marine environment, as well.

Although the GR curve in this case was very oscillatory, there is another application of the same workflow (Lorena Gonçalves André et al. (presented in this congress), where the GR and LLD curves were well behaved. In that case, the phase 3 and 4 were not necessary.

As for the P\_90, P\_50 and P\_10 curves, we verify that the S1s\_P\_90 curve was more able to reproduces and follow the pattern induced by the discrete analytic S1 values.

A resume code for Python Implementation is shown in the Appendix A, and the complete code can be accessed in this github:

<https://github.com/EdsonAlonso/Synthetic-S1-python-implementation.git>

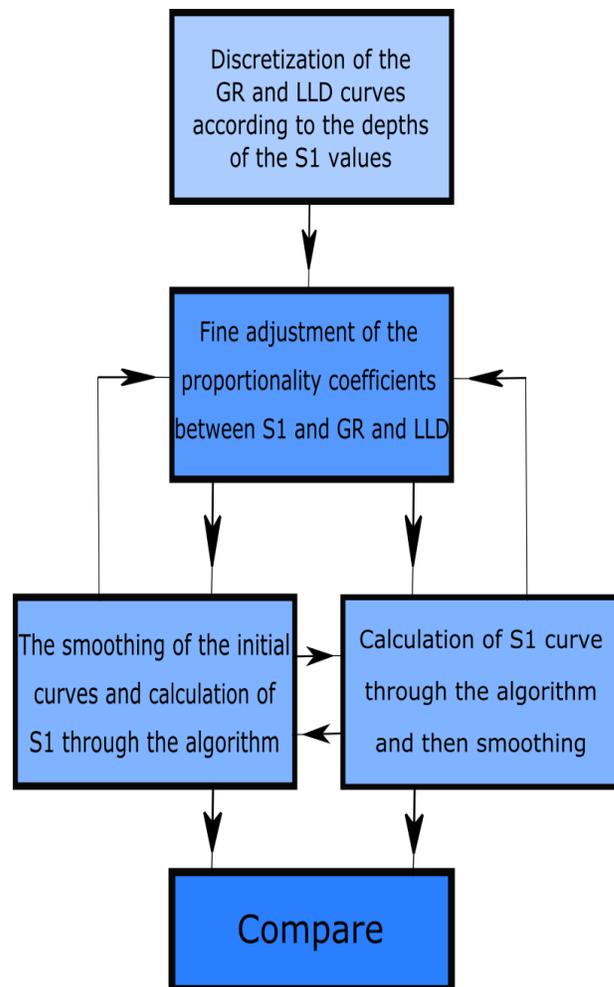


Figure 4: Workflow proposed in this study.

## Results and Discussion

In this section will be made a qualitative interpretation of the products generated by the algorithm proposed by Lima e Silva (personal communication).

Figure 5 shows the discretization of the GR and LLD curves of the well 4-BRSA-420-ESS in relation to the depths of the analytical S1 measured in Rock-Eval pyrolysis. There are 122 samples collected between 2300-2900m. The vertical continuity of the data is observed up to 2750m. Below this value there are 150m interval without samples and this affects the adjustment power of the algorithm. It is worth remembering that the geochemical data were measured by cutting samples having uncertainty as to their estimated depth and as to the determination of the Rock-Eval pyrolysis products. On the other hand, cuttings represent an average of 3m interval, and logs have 30cm sampling interval, 10 times denser.

Figure 6 shows the final curve related to the synthetic S1s\_P90 curve. As we said previously, the P\_90 curve behaved much better than P\_50 and P\_10 curves.

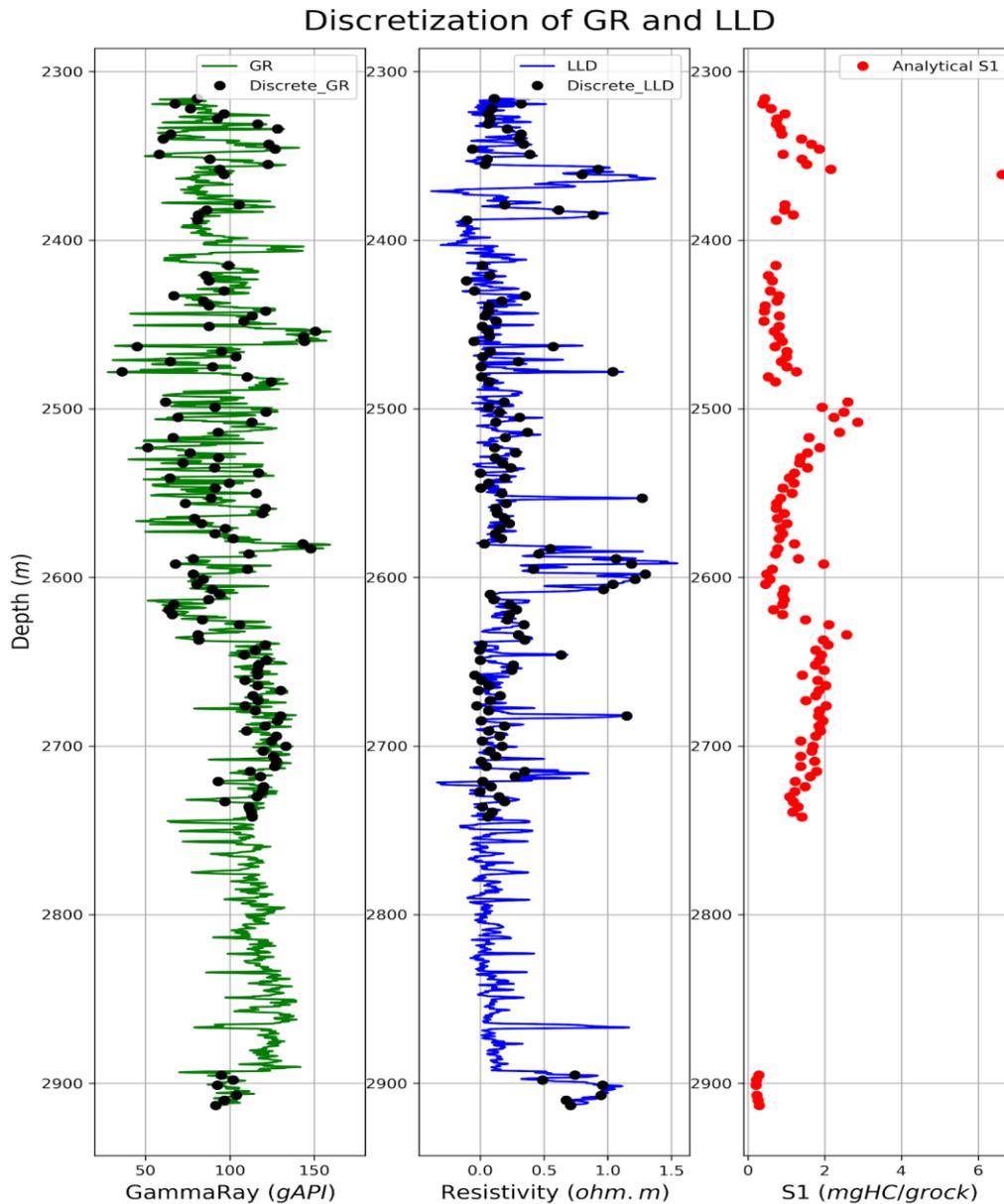
### Free hydrocarbon content (S1) from well logs

It is important to mention the pattern existing between both the synthetic curve and the analytical measurements. In fact, the two intervals for which we would like to draw attention are about 2630/2900 meters. It is clear that the resulting curve follows the behavior of the analytical data. The existence of an outlier does not seem to damage the final result, as we can see in Figure 6.

The result shown in Figure 6 means a new to extract information about geochemical properties of our data in a cheaper way, because the process of calculating the

analytical value of S1 via pyrolysis is expensive. On the other hand, well logs are not only necessary, but the curves necessary for the S1s algorithm and workflow application are the most basic ones.

Finally, we would like to add one more argument for the validation of the S1s algorithm and our workflow. Note that there is no data in the interval 2750/2880 m and the S1s curve could provide a S1s curve to complement the analytical data. Anyway, the final curve keeps its behavior and trends, showing the strength of the method.



**Figure 5:** Discretization of the GR and LLD curves according with the S1 measurements depths.

### SYNTHETIC S1 SMOOTHED

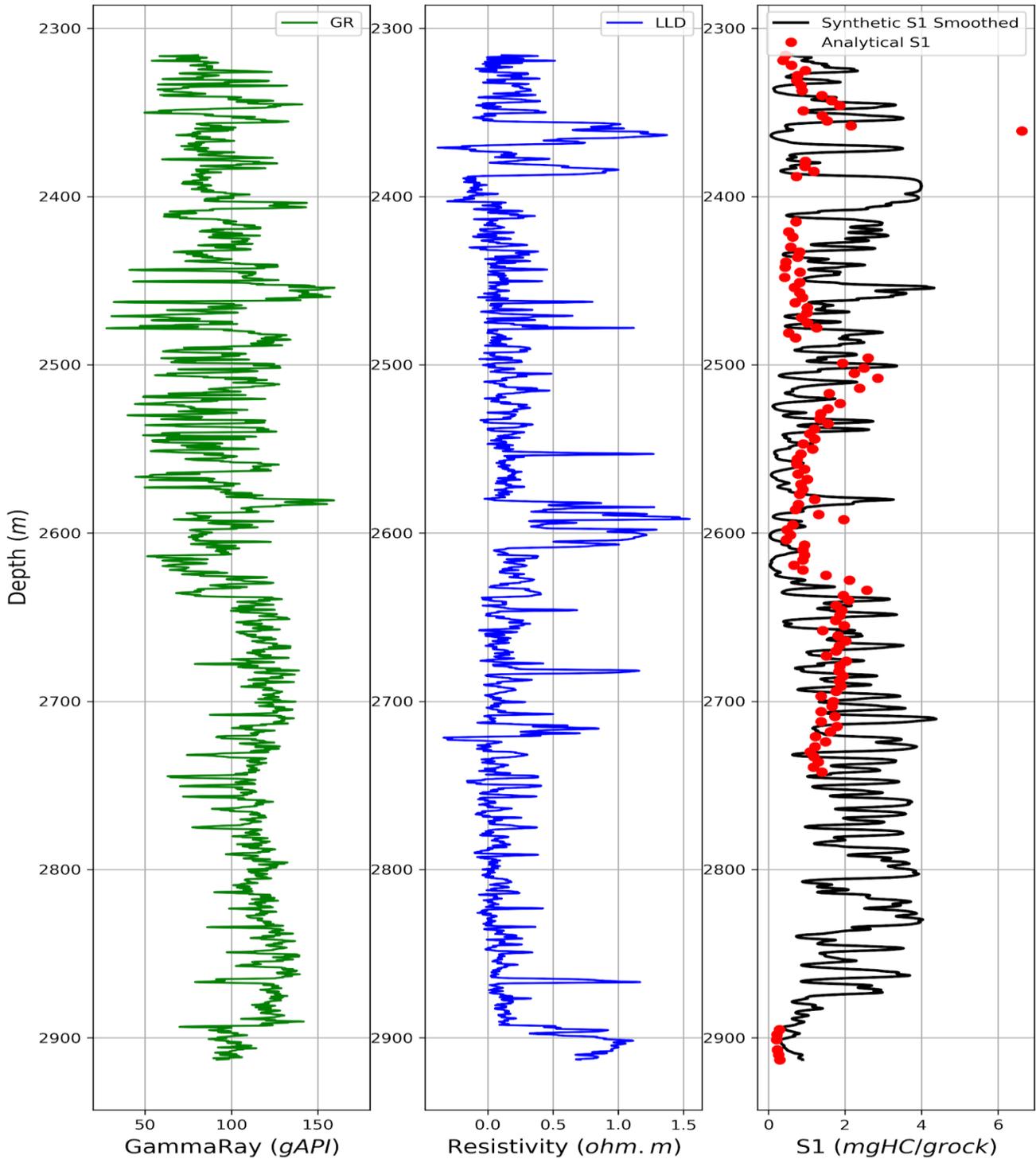


Figure 6: Plot of the GR, LLD, and S1s\_P90 curves.

## Free hydrocarbon content (S1) from well logs

### Preliminary Conclusions

The adjustment of the S1s\_P90 smoothed curve shows really satisfactory because a few mathematical tools were needed. The method is proposed in a marine section but it should, and will, be applied in other places in order to expand and verify it.

The amount of Rock-Eval pyrolysis data should have a good vertical continuity to ensure the effectiveness of the method. We suggest the application of new signal filters, keeping in mind that the less the qualitative information in the data in change, the better.

### Acknowledgments

To BDEP/ANP for providing data to perform this work and to Federal Fluminense University for the technical support and permission to use computer labs.

### References

Jarvie, D.M. (2012). **Shale Resource Systems for Oil and gas: Part 1 – Shale-gas resource Systems**. in J.A. Breyer, ed., *Shale Reservoirs – Giant resources for the 21<sup>st</sup> century*: AAPG Memoir 97, p. 69-87.

Jarvie, D.M. (2012). **Shale Resource Systems for Oil and gas: Part 2 – Shale-oil resource Systems**. in J.A. Breyer, ed., *Shale Reservoirs – Giant resources for the 21<sup>st</sup> century*: AAPG Memoir 97, p. 89-119.

Passey, Q.R.; Creaney, S.; Kulla, J.B.; Moretti, F.J.; Stroud, J.D. (1990). **A practical model for organic richness from porosity and resistivity logs**. AAPG Bulletin, v.74 (12), p. 1777-1794.

Winter, W.R.; Jahnert, R.J.; França, A.B. (2007). **Bacia de Campos**. Boletim de Geociências da Petrobras, v.15 (2), p. 511-529.

Pereira, G.B. (2015). **Bacia de Campos - Sumário Geológico e Setores em Oferta**. 13<sup>a</sup> Rodada de Licitações de Petróleo e Gás. Superintendência de Definição de Blocos SDB, ANP.

### Appendix A : Python Implementation

```
def S1_sint_p90 (well, header, nrows, cols): "
```

This function receives a .txt file corresponding to some well log data, its header, the number of rows needed to be spiked for the start of the numerical data, and the columns corresponding to the Gamma Ray (GR) and Resistivity data (LLD), in this order.

Input -

Well -> .txt file (can be also csv file, just remember to use the correct suffix) - example: well = Y-BR-YYY-PRF.txt

header -> String vector containing the names of the columns that will be used - example : ['DEPTH', 'GR', 'LLD'] (REMEMBER THE ORDER)

nrows -> Integer value meaning the number of rows that the program must no skip in order to find numerical data.

The program will not work if it finds string instead of float data.

cols -> Array containing the columns indexes. Remember that for the correct usage of the algorithm the GR, and LLD columns are needed. Put here as the following example : If GR is the second column, and LLD is the third, cols = [1,2]

Output-

Array of the sum of all functions that is equal to the value of synthetic S1\_P90.

```
""
curva = pd.read_csv('well', sep = '\s+', skiprows = nrows, usecols
= cols, names = header)
Gamma = curva.header[2]
LLD = curva.header[3]
GR = np.array(Gamma)
RT = np.array(LLD)
f1 = np.zeros(len(GR))
f2 = np.zeros(len(GR))
f3 = np.zeros(len(GR))
f4 = np.zeros(len(GR))
f5 = np.zeros(len(GR))
f6 = np.zeros(len(GR))
f7 = np.zeros(len(GR))
```

```
for i in range(len(GR)):
    if RT[i]>=4.0 and GR[i]>=90.0:
        f1[i] = (1.6*GR[i]**3)/(13399.221*RT[i]**2)

    elif RT[i]>=4.0 and GR[i]<90.0:
        f2[i] = (0.5*GR[i]**3)/(13399.221*RT[i]**2)

    elif RT[i]>=2.0 and RT[i]<4.0 and GR[i]>=90.0:
        f3[i] = (1.6*GR[i]**3)/(13399.221*RT[i]**4)

    elif RT[i]>=2.0 and RT[i]<4.0 and GR[i]<90.0:
        f4[i] = (0.7*GR[i]**3)/(13399.221*RT[i]**4)

    elif RT[i]>=1.036 and RT[i]<2.0 and GR[i]>=90.0:
        f5[i] = (1.6*GR[i]**2)/(800.0*RT[i]**8)

    elif RT[i]>=1.036 and RT[i]<2.0 and GR[i]<90.0:
        f6[i] = (GR[i]**2)/(800.0*RT[i]**8)

    elif RT[i]<1.036:
        f7[i] = 15.0
    else:
        f1[i] = f2[i] = f3[i] = f4[i] = f5[i] = f6[i] = f7[i] = 0.0
```

```
return f1 + f2 + f3 + f4 + f5 + f6 + f7
```

Complete codes used in this resume are in this github:

<https://github.com/EdsonAlonso/Synthetic-S1-python-implementation.git> (