UNIVERSIDADE DE BRASÍLIA (UnB)

INSTITUTO DE QUÍMICA

QUÍMICA TECNOLÓGICA

HUGO DA SILVA ROCHA

IMPLEMENTAÇÃO DA ANÁLISE DE COMPONENTES PRINCIPAIS (PCA) AO SOFTWARE GENERAL RMN ANALYSIS TOOLBOX (GNAT) - APLICAÇÃO NA DIFERENCIAÇÃO DE ÓLEO DE OLIVA, SOJA E CANOLA.

Brasília – DF

2022

HUGO DA SILVA ROCHA

IMPLEMENTAÇÃO DA ANÁLISE DE COMPONENTES PRINCIPAIS (PCA) AO SOFTWARE GENERAL RMN ANALYSIS TOOLBOX (GNAT) - APLICAÇÃO NA DIFERENCIÇÃO DE ÓLEO DE OLIVA, SOJA E CANOLA.

Trabalho de conclusão de curso apresentado ao Instituto de Química da Universidade de Brasília, como requisito parcial para a obtenção do título de Bacharel em Química Tecnológica.

Orientador: Dr. Jez William Batista Braga

Coorientador: Dr. Mathias Nilsson

Brasília – DF

2022

AGRADECIMENTOS

Agradeço primeiramente à minha família, que me incentivaram em diversas etapas na minha vida, mesmo nos momentos mais difíceis, a continuar seguindo meus objetivos.

Aos professores e amigos da Universidade de Brasília, que proporcionaram oportunidades para diversas alegrias compartilhadas e desafios superados.

Nominalmente agradeço aos amigos mais próximos que me aconselharam e apoiaram nessa trajetória, sendo eles Daniele, Calil, Samia, Caio, Rafaela, Thalita, Mauro e Simone.

Ao professor Cláudio Francisco Tormena, da Universidade de Campinas, por disponibilizar o laboratório para aquisição dos dados e pelas aulas sobre ressonância magnética.

Ao professor Mathias Nilsson e Dr. Guilherme F. D. Poggetto, da University of Manchester, pelo meu recebimento como aluno à distância em seu trabalho e pelas várias horas disponibilizadas para aulas sobre programação.

Ao professor Jez, por ter sido a principal influência na área que escolhi prosseguir com meus estudos e pelo apoio no meu crescimento nessa área.

A Dr. Tereza Pastore, por ter me proporcionado inúmeras oportunidades e aprendizados no inicio da minha carreira. Além de uma excelente orientadora uma grande amiga.

À Universidade de Brasília e a Universidade de Campinas por manter e disponibilizar os instrumentos necessários para este trabalho.

Aos órgãos de fomento INCTBio, CNPq, FINEP, e CAPES pelo suporte financeiro para o desenvolvimento desse trabalho

RESUMO

O uso de uma Interface Gráfica de Usuário (GUI) vem assumindo um papel importante como ferramenta visual para facilitar a interação do usuário com ferramentas matemáticas, como por exemplo, o pré-processamento de um conjunto de espectros de RMN, eliminando a necessidade dos usuários aprenderem uma linguagem de programação ou digitarem comandos para a obtenção de resultados. Os experimentos realizados em RMN de 1H e 13C têm o potencial de fornecer informações que caracterizam a composição do conjunto de amostras analisadas. Contudo, dependendo dos objetivos do estudo, esse conjunto de dados necessita de ferramentas de análise multivariada para extração dessas informações, sendo muitas vezes inevitável a importação e tratamento dos dados em diversos pacotes de softwares para uma análise completa dos espectros (pré-processamentos + analises multivariadas), em parte devido à falta de um software de processamento eficiente e que reúna as ferramentas necessárias para a análise de dados de RMN. Uma ferramenta gráfica de código aberto implementada no ambiente do MATLAB é o GNAT (General NMR Analysis Toolbox). Este software gratuito se trata de uma GUI desenvolvida para processamento básico de dados de RMN (por exemplo, transformada de Fourier, correção de linha de base, correção de fase), bem como técnicas mais avançadas (por exemplo, deconvolução de referência e ferramentas de análise DOSY e SCORE). O objetivo principal desse trabalho foi desenvolver o código para a implementação da função PCA ao pacote computacional GNAT, visando contribuir para o desenvolvimento de uma ferramenta mais geral para análise de dados de RMN. Espectros de RMN de ^{1}H e ^{13}C de óleos de soja, canola e azeites foram obtidos na UNICAMP visando testar as funcionalidades das funções desenvolvidos nesse programa. A validação dos cálculos quimiométricos desenvolvidos no GNAT foi realizada comparando os resultados do modelo PCA com os espectros de RMN processados pelo software pago PLS_toolbox. Os resultados mostram que o modulo de funções desenvolvidos no GNAT apresentaram o mesmo valor de variância explicada para o conjunto de dados analisados no PLS_toolbox. Em ambos os programas, o conjunto de amostras de óleo de azeite foi separado do conjunto de óleo de soja e canola, sendo que no GNAT foi possível realizar a otimização dos parâmetros de pré-processamento imediatamente seguida da análise quimiométrica em todos os testes de otimização do programa. A perspectiva futura desse trabalho é desenvolver outras ferramentas de análise multivariada com foco em espectros de RMN, visando fornecer ao usuário o acesso a diversas opções para analise espectral de seus dados.

Palavras-chave: RMN, PCA, MATLAB, Óleos Vegetais

ABSTRACT

The use of a Graphical User Interface (GUI) has assumed an important role as a visual tool to facilitate user interaction with mathematical tools, such as the pre-processing of a set of NMR spectra, eliminating the need for users to learn a programming language or type commands to get results. The experiments carried out in 1H and 13C NMR have the potential to provide information that characterizes the composition of the set of analysed samples. However, depending on the objectives of the study, this data set needs multivariate analysis tools to extract this information, and it is often unavoidable to import and process the data in different software packages for a complete analysis of the spectra (pre-processing + analysis multivariate), in part due to the lack of efficient processing software that gathers the necessary tools for the analysis of NMR data. An open source graphical tool implemented in the MATLAB environment is the GNAT (General NMR Analysis Toolbox). This free software is a GUI designed for basic NMR data processing (e.g. Fourier transform, baseline correction, phase correction) as well as more advanced techniques (e.g. reference deconvolution and DOSY and SCORE analysis). The main objective of this work was to develop the code for the implementation of the PCA function to the GNAT computational package, aiming to contribute to the development of a more general tool for analysing NMR data. 1H and 13C NMR spectra of soybean, canola and olive oils were obtained at UNICAMP in order to test the functionalities of the functions developed in this program. The validation of the chemometric calculations developed in GNAT was performed by comparing the results of the PCA model with the NMR spectra processed by the paid software PLS_toolbox. The results show that the module of functions developed in GNAT presented the same value of explained variance for the dataset analysed in PLS_toolbox. In both programs, the set of olive oil samples was separated from the set of soybean and canola oil, and in GNAT it was possible to optimize the pre-processing parameters immediately followed by the chemometric analysis in all optimization tests. from the program. The future perspective of this work is to develop other multivariate analysis tools focused on NMR spectra, aiming to provide the user with access to several options for spectral analysis of their data.

Keywords: NMR, PCA, MATLAB, Vegetable Oils

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS (Ordem alfabética)

GUI - Interface Gráfica de Usuário (*Graphical User Interface).*

GNAT - Toolbox geral de análise de RMN (*General RMN Analysis Toolbox*).

RMN – Ressonância magnética Nuclear (*Nuclear Magnetic Ressonance*)

PC – Componente Principal (*Principal Component*)

PCA – Análise dos Componentes Principais (*Principal Component Analysis*)

SUMÁRIO

1 INTRODUÇÃO

A espectroscopia de Ressonância Magnética Nuclear (RMN) se trata de uma técnica que permite observar fenômenos físicos e químicos que geram um espectro rico em informações, podendo este ser considerado uma impressão digital única dos núcleos ativos presentes nas amostras (ALONSO-SALCES; HOLLAND; GUILLOU, 2011). Espectros de RMN de 1H possuem como vantagem alta abundância natural dos isótopos detectáveis de hidrogênio e alto momento magnético dos núcleos ^{1}H, o que permite a aquisição de espectros de ^{1}H RMN com uma maior relação sinal/ruído em um menor tempo de aquisição. (BARISON *et al.*, 2010)**.** Em contrapartida, os espectros de RMN de 13C têm uma janela espectral mais ampla quando comparada aos espectros de 1H, apresentando consequentementeespectros que raramente apresentam picos de carbono sobrepostos, permitindo os sinais prontamente interpretáveis. Contudo, as análises exigem amostras mais concentradas e, consequentemente, longos tempos de aquisição para que a análise seja concluída (LIA *et al.*, 2020)

A análise de dados espectrais de RMN traz inúmeros desafios devido à complexidade dos sinais e fatores externos que podem comprometer a reprodutibilidade dos experimentos, e.g. instabilidades instrumentais, pH, força iônica, temperatura, entre outros, podendo levar a interpretações incorretas dos resultados (ANDERSON *et al.*, 2011; FORSHED *et al.*, 2005). Por esse motivo, técnicas de reconhecimento de padrões baseados em análise multivariada de dados, como por exemplo PCA (do inglês, Principal Component Analysis), podem ser requeridas para extração das informações necessárias para interpretação dos dados, detecção de adulteração e descrição de perfis metabólicos (DAOLIO *et al.*, 2008; LENZ *et al.*, 2003; SOUSA; MAGALHÃES; FERREIRA, 2013).

Uma proposta apresentada por vários autores é utilizar a espectroscopia de RMN como meio para a caracterização de alimentos, como por exemplo os azeites (FAUHL; RENIERO; GUILLOU, 2000; KEIFER, 2003). O RMN ^{1}H de óleos comestíveis fornece informações importante para esse tipo de amostra, como: classes lipídicas, nível de insaturação, frações molares de ácidos graxos e sinais de compostos minoritários, como esteróis, esqualeno e terpenos (FAUHL; RENIERO; GUILLOU, 2000; MANNINA; SOBOLEV; A., 2003). A detecção dessas variáveis possibilita realizar o controle de qualidade de óleos comestíveis de acordo com a sua composição, grau de oxidação e determinar a autenticidade ou adulteração

dessas amostras. (CICHELLI; PERTESANA, 2004; HÉBERGER; RAJKÓ, 2002; LUCASIUS; KATEMAN, 1993).

Um desafio comum para essas amostras como azeites e óleos vegetais é a identificação geográfica de origem desse tipo de produto. O potencial da estatística multivariada como ferramenta promissora para autenticar e classificar produtos alimentícios de acordo com sua origem geográfica já foi apresentado na literatura e revisado criticamente por diversos estudos (REZZI *et al.*, 2005; TZOUROS; ARVANITOYANNIS, 2010)**.** Como exemplo, CHRISTY *et al.* (2004), utilizando análise de componentes principais (PCA), regressão de mínimos quadrados parciais (PLS) e métodos aplicados para pré-tratamentos de dados, determinaram a composição de ácidos graxos de óleos vegetais comestíveis, permitindo assim detectar e quantificar a adulteração em azeite por espectroscopia de infravermelho próximo e usando técnicas quimiométricas.

Da mesma forma, diferentes ferramentas quimiométricas, como PLS, PLS-DA, OPLS-DA, etc, foram comparadas na literatura buscando estabelecer os parâmetros otimizados para uma variedade de amostras (BRESCIA *et al.*, 2003; CHRISTY *et al.*, 2004; CICHELLI; PERTESANA, 2004; FAUHL; RENIERO; GUILLOU, 2000; JAKAB; HÉBERGER; FORGÁCS, 2002; OLIVERI *et al.*, 2011)**.**

Uma variedade de softwares estão disponíveis para realização de analises estatísticas em dados de RMN de alta resolução, assim como de demais espectros. Esses softwares são comumente disponíveis por diferentes meios: fornecidos por fabricantes de espectrômetros, (BRUKER, 2021), pacotes comerciais,(GÜNTERT *et al.*, 1992; MAGRITEK, 2021) ou softwares livres. (COBAS; SARDINA, 2003; ACD/LABS, 2022). Os softwares comerciais disponíveis para análise multivariada, tanto online quanto disponíveis para desktop, como por exemplo: Metabolomics, SIMCA, PLS Toolbox, The Unscrambler, etc., tendem a ser caros e fornecem poucas funcionalidades que sejam específicas para uma determinada técnica instrumental, o que muitas vezes faz com que as melhores condições de correção de linha de base, zero-filling, alinhamento dos sinais espectrais, etc., não sejam encontradas em um único software.

Dentre os pacotes de processamento espectral de RMN disponíveis de forma gratuita se destaca o GNAT (do inglês, General RMN Analysis Toolbox). Este programa se trata de um pacote de software desenvolvido pelo laboratório Manchester NMR Methodology (University of Manchester), dentro do ambiente MATLAB®, destinado ao processamento de conjuntos de

dados RMN adquiridos em experimentos de difusão, relaxamento e cinética. O processamento básico de dados de RMN é fornecido dentro do próprio programa, assim como técnicas avançadas como a deconvolução de referência e reconstrução FID de deslocamento puro. Contudo, o software ainda conta com poucas ferramentas de análise multivariada desses dados, sendo essa uma das principais vertentes para seu desenvolvimento. (CASTAÑAR *et al.*, 2018).

2 OBJETIVOS

2.1 Objetivos Gerais

Desenvolver, dentro do ambiente do MATLAB, o código para a implementação da função PCA ao pacote computacional GNAT e testar suas funcionalidades para análise de dados espectrais de RMN.

2.2 Objetivos Específicos

• Implementar as funções para realização de alguns dos principais pré-processamentos de dados e PCA dentro da plataforma gráfica. Como exemplo, centralização dos dados na média e binning.

• Realizar as eventuais modificações no código do programa para melhor funcionamento da GUI, tanto no aspecto visual quando na otimização dos cálculos matriciais.

• Realizar a obtenção dos dados espectrais de amostras de óleo de oliva, soja e canola comercialmente disponíveis.

• Analisar os espectros de RMN de ^{1}H e ^{13}C dos óleos vegetais visando demonstrar a implementação do PCA no GNAT, assim como avaliar as funcionalidades dos softwares desenvolvidos nesse processo.

3 REVISÃO BIBLIOGRÁFICA

3.1 MATLAB

O MATLAB (do inglês MATrix LABoratory) é um software de alta performance que possibilita a construção de funções e rotinas de cálculos matemáticos em uma linguagem de programação similar às linguagens tradicionais como Fortran, C/C++, Visual Basic, etc. Contudo, o software se diferencia dos demais por funções internas que permitem uma visualização gráfica dos resultados mais apresentável que as demais linguagens. (HUNT, LIPSMAN, ROSENBERG, 2001). Além disso, o MATLAB possui toolboxes para aplicações automatizadas para rotinas de funções comumente utilizadas em operações que envolvem regressões, machine learning e quimiometria, como por exemplo: Statistics and Machine Learning Toolbox™, Curve Fitting Toolbox™, Deep Learning Toolbox™ e PLS_Toolbox™. (MATLAB & SIMULINK, 2022; EIGENVECTOR, 2022)

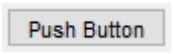
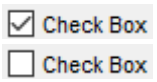
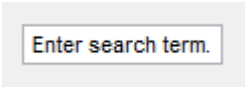
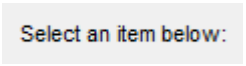
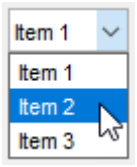
Uma Interface Gráfica de Usuário, (do ingês Graphic User Interface,GUI) é uma exibição gráfica executada pelo MATLAB, ao qual utiliza objetos gráficos como botões, imagens e caixas de texto, como meio de interação entre o usuário e as rotinas de funções utilizadas em sua construção. Essa estrutura elimina a necessidade de interação com a linguagem de programação por parte do usuário.

É possível escolher entre duas metodologias para a criação de um aplicativo/GUI no MATLAB:

- App Designer e GUIDE: quando o usuário deseja criar um aplicativo em um ambiente de arrastar e soltar os objetos gráficos para criar a interface do usuário, sem a necessidade de utilizar linhas de código para isso;
- Programaticamente: quando o usuário deseja criar a interface do usuário de um aplicativo escrevendo as funções e estruturas gráficas do início.

Desenvolver uma GUI por programação implica em descrever todos os componentes, suas propriedades, respostas a erros e comportamento por meio de linhas de código. Nessa abordagem, você cria uma figura para servir como “contêiner” para sua interface de usuário e adiciona os objetos gráficos e funções a ela programaticamente. Os objetos gráficos permitem transferir o controle da execução das funções para o usuário, sendo esses apresentados na Tabela 1:

Tabela 1. Exemplo da estrutura dos controles disponíveis para interação do usuário com as funções presentes no algoritmo da GUI.

‘pushbutton'		Botão para execução de funções após click do botão do mouse.
'checkbox'		Opção de seleção de condições múltiplas de um grupo.
'radiobutton'		Opção de seleção de condições única de um grupo.
'edit'		Campo de texto editável
'text'		Campo de texto estático
'popupmenu'		Menu pop-up (também conhecido como menu suspenso) que se expande para exibir uma lista de opções.

A estrutura final de uma GUI – independente da metodologia utilizada para sua construção – pode ser compartilhada para uso no MATLAB em versões que apresentem compatibilidade com as funções presentes na GUI, assim como aplicativos compilados autônomos para desktop ou web. Informações sobre a estrutura utilizada na construção da GUI desenvolvida no trabalho pode ser visualizado nos Anexos 1.1 até 1.3.

3.2 GNAT

O GNAT se trata de uma GUI de código aberto para analises e tratamento de espectros de RMN. Esta se encontra disponível gratuitamente sob a GNU General Public License, onde todo o código fonte está escrito na linguagem MATLAB, disponível para edição e compatível com a versão R2015a e superior. Versões compiladas do Toolbox que são executadas independentemente de qualquer instalação do MATLAB estão disponíveis para Windows, Mac e Linux.

A interface gráfica do usuário consiste em uma janela principal, a partir da qual está disponível o acesso à maioria dos recursos de processamento e análise (Figura 1).

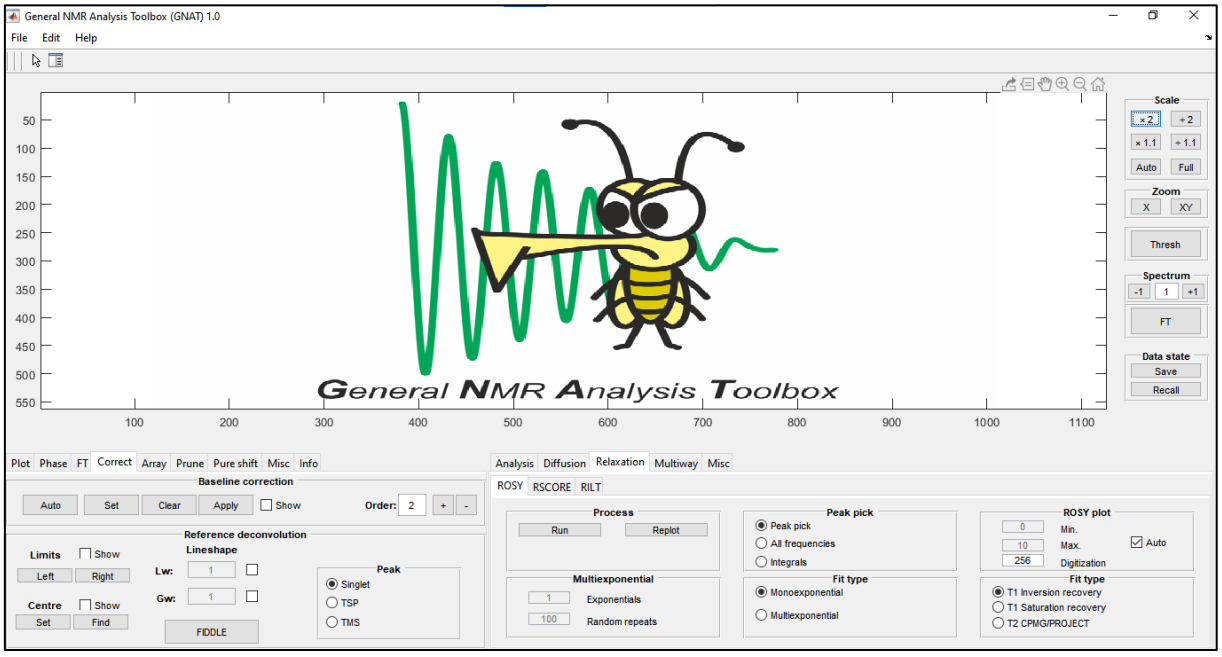


Figura 1. Estrutura inicial do GNAT após a inicialização no MATLAB

A GUI é dividida entre processamento e recursos básicos no grupo de guias à esquerda (e.g. transformada de Fourier, controle de plotagem e correção de linha de base) e análises mais avançadas (e.g., DOSY, ROSY e RSCORE) no grupo de guias à direita. (Figura 1). Versões revisadas mais atuais do programa podem ser baixadas no site https://www.nmr.chemistry.manchester.ac.uk/?q=node/29 , o qual possui suporte contínuo por meio de novas implementações, melhorias e correções de bugs.

3.3 Quimiometria

3.3.1 PCA

O PCA é uma técnica poderosa para extração de informação em grandes conjuntos de dados. Ela tem como objetivo reduzir um conjunto maior de variáveis preditoras a um conjunto menor com perda mínima de informações, por meio da combinação linear das variáveis originais para a formação de novas variáveis, chamadas de componentes principais (CPs), que maximizam a variância explicada para um determinado número de componentes escolhido.

Seja a matriz de dados $\mathbf{X}$ de tamanho $n \times m$, onde n é o número de amostras e m é o número de variáveis. Supondo que $\mathbf{X}$ esteja centralizado, ou seja, as médias das colunas foram

subtraídas e agora são iguais a zero. A matriz de covariância $m \times m$ (**C)** é dada por $\mathbf{C} = \mathbf{X^T X}/(n-1)$. É uma matriz simétrica e, portanto, passível de ser diagonalizada:

$$\mathbf{C} = \mathbf{V L V^T} \tag{1}$$

onde **V** é uma matriz de autovetores (onde cada coluna é um autovetor) e **L** é uma matriz diagonal com autovalores $\boldsymbol{\lambda_i}$ organizada em ordem decrescente.

Os autovetores são chamados de eixos principais, já as projeções dos dados nos eixos/componentes principais são chamadas de escores. Os escores da *j*-ésima componente principal é dada pela *j*-ésima coluna de **XV**. As coordenadas do *i*-ésimo ponto de dados no novo espaço PC são dadas pela i-ésima linha de **XV**.

É possível realizar a decomposição da matriz **X** aplicando a decomposição de valor singular, (SVD, do inglês *Singular Value Decomposition*), por meio da formula:

$$\mathbf{X} = \mathbf{U S V^T} \tag{2}$$

onde **U** é uma matriz unitária e **S** é a matriz diagonal de valores singulares $\boldsymbol{s_i}$. A partir da decomposição de **X**, é possível obter a seguinte relação da multiplicação das matrizes:

$$\boldsymbol{C} = \frac{\mathbf{V S U^T U S V^T}}{n-1} = \mathbf{V}\frac{\boldsymbol{S^2}}{n-1}\mathbf{V^T} \tag{3}$$

significando que os vetores singulares **V** são direções principais e que os valores singulares estão relacionados aos autovalores da matriz de covariância via $\lambda_i = \frac{s_i^2}{(n-1)}$. Os escores das componentes principais são dados por

$$\mathbf{XV} = \mathbf{USV^TV} = \mathbf{US} \tag{4}$$

Sumarizando os pontos principais desse processo, tem-se que:

1. As colunas da matriz **US** são chamadas de escores (*scores*), comumente representadas por **T** em diversos artigos de textos de quimiometria;
2. Autovalores λ_i mostram variâncias dos respectivos PCs;
3. Os valores de escores padronizadas são dados por colunas de **U** e os pesos (*loadings*) são dados pelas colunas de **V**.
4. Os valores de *escores* e *pesos* na forma apresentada acima está correto somente se **X** estiver centralizada na média.

5. Os cálculos de decomposição de **X** consideram que a estrutura dessa matriz possui amostras nas linhas e variáveis nas colunas.
6. Para reduzir a dimensionalidade dos dados de *m* para um rank ou posto menor $k<m$, é necessário selecionar as *k* primeiras colunas de **U** e a matriz superior $k \times k$ de **S**. O produto $\mathbf{U_k S_k}$ é a matriz **T** ($n \times k$) contendo os primeiros *k* componentes principais.
7. A matriz **U** tem as dimensões de $n \times n$ e **V** possui dimensões $m \times m$. Para os casos onde $n>m$, as últimas $n - m$ colunas de **U** não são necessárias no cálculo das componentes principais; deve-se, portanto, usar o algoritmo SVD de tamanho econômico (*economy size*) que retorne **U** de tamanho $n \times m$. (Figura 2). Essa remoção diminui a exigência de poder computacional no cálculo das componentes principais.

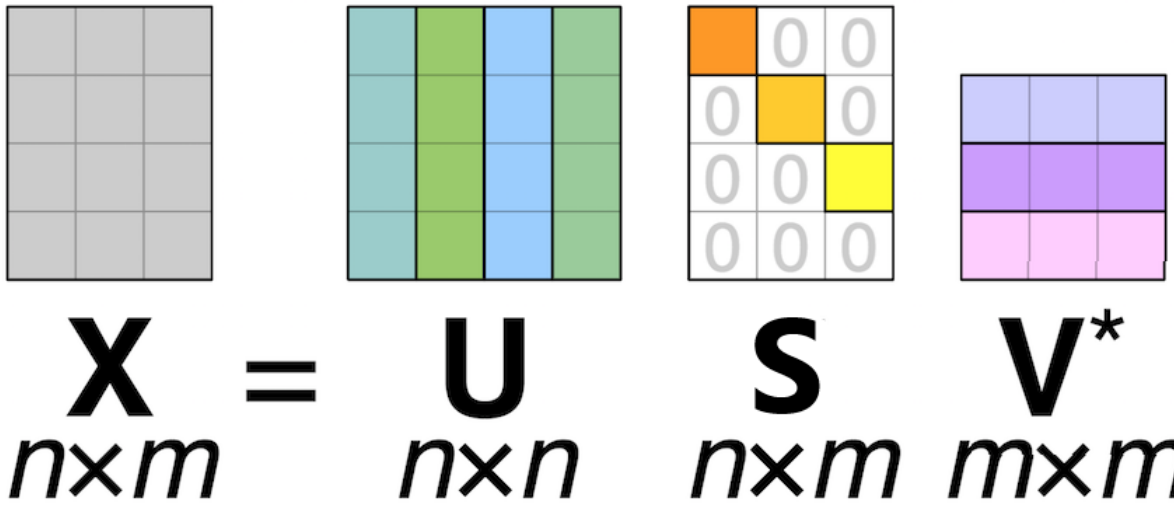


Figura 2. Componentes da decomposição de valor singular de uma matriz X.

A variância explicada obtida na decomposição da matriz **X** aparece em ordem decrescente em cada componente principal (PC, *Principal Components*), sendo geralmente apresentada em percentual. A primeira PC busca descrever a direção da maior variação no conjunto de dados. Já a segunda PC, ortogonal à primeira, descreve a direção da segunda maior variação e assim por diante. Dessa forma, cada nova PC representa uma porção menor da variância explicada. Nos casos em que a porcentagem de variância explicada apresentada na primeira componente principal é muito baixa, uma segunda componente principal deve ser considerada para descrever **X**. As PCs com porcentagem de variância explicada da ordem do ruído presente no conjunto de dados geralmente não são significativas (HARROU et al., 2015), e dessa forma, desconsideradas na modelagem. Dessa forma, verifica-se que os dados podem ser adequadamente descritos utilizando somente informações importantes na sua caracterização.

3.3.2 Outliers

Os *outliers* representam as amostras que possuem um comportamento distinto em comparação com outras amostras do mesmo conjunto de dados. A presença dessas amostras interfere no cálculo da PCA, prejudicando gravemente a interpretação da correlação dos dados. Assim, a detecção de outliers é necessária e visa reduzir a presença de variação desnecessária no conjunto de dados.

Os Resíduos Q e o T^2 de Hotelling são ferramentas comuns na identificação dos outliers. A estatística T^2 é uma medida da variação do modelo PCA, já a estatística Q é uma medida da quantidade de variação não capturada pelo modelo PCA, apresentada em sua matriz de resíduos **E** (MUJICA *et al.*, 2011). A estatística T^2 é definida pela distância de Mahalanobis enquanto a estatística Q é definida pela distância euclidiana (KOURTI; MACGREGOR, 1995; MACGREGOR; KOURTI, 1995; QIN, 2003)

3.3.2.1 Resíduos Q

A estatística Q mede a projeção ortogonal de uma amostra ao espaço definido pelo modelo PCA. (HARROU *et al.*, 2015). Em outras palavras, a matriz resultante nesse cálculo pode ser vista como uma medida do quanto a amostra em questão é bem modelada. Seu cálculo pode ser expresso como:

$$\mathbf{Q} = \mathbf{e_i e_i^T} = \mathbf{x_i}\left(\mathbf{I} - \mathbf{P_k P_k^T}\right)\mathbf{x_i^T} \tag{5}$$

Onde $\mathbf{e_i}\ \mathbf{e_i}^{\mathbf{T}}$ é a *i*-ésima linha de **E**; $\mathbf{P_k}\,\mathbf{P_k}^{\mathbf{T}}$ é a matriz dos *k* vetores de pesos retidos no modelo, onde cada vetor é uma coluna de $\mathbf{P_k}\,\mathbf{P_k}^{\mathbf{T}}$ e **I** é a matriz identidade com mesmo tamanho de $\mathbf{P_k}\,\mathbf{P_k}^{\mathbf{T}}$.

Os limites de confiança para Q podem ser calculados utilizando os autovalores da matriz de covariância de **X (**λ_i**)** obtidos pelas equações: (JACKSON; MUDHOLKAR, 1979)

$$Q_\alpha = \Theta_1\left(\frac{C_\alpha\sqrt{2\Theta_2 h_0^2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0(1-h_0)}{\Theta_1}\right) \tag{6}$$

$$\boldsymbol{\Theta_i} = \sum_{j=k+1}^{p} \boldsymbol{\lambda_j^i} \tag{7}$$

$$\boldsymbol{h_0} = \mathbf{1} - \frac{\mathbf{2\Theta_1\Theta_3}}{\mathbf{3\Theta_2^2}} \tag{8}$$

Onde **Cα** é o desvio normal correspondente ao percentil superior **(1 − α)**

Esses limites de confiança são calculados com base nas suposições de que as medições são independentes do tempo e possuem uma distribuição normal. As amostras distribuídas fora dos limites de confiança são consideradas anômalas (outliers) com base nos dados usados para construção do modelo PCA. (JACKSON; MUDHOLKAR, 1979)

3.3.2.2 T^2 de Hotelling

Como dito anteriormente, o T^2 de Hotelling pode ser definido pela distância de Mahalanobis no espaço do modelo PCA. A distância de Mahalanobis possibilita descrever a variação na distribuição amostral para diferentes planos de projeção de dados considerando sua importância para o modelo. Dessa forma, é possível verificar que a distância de distribuição amostral em algumas direções é mais significativa do que a distância em outras direções (Figura 3).

Essas distâncias podem fornecer uma medida da variação em cada amostra dentro do modelo PCA. Assim, as amostras com alta variações podem ser facilmente identificadas e consideradas como outliers, como por exemplo a amostra em verde presente na figura 2. (HOTELLING, 1933):

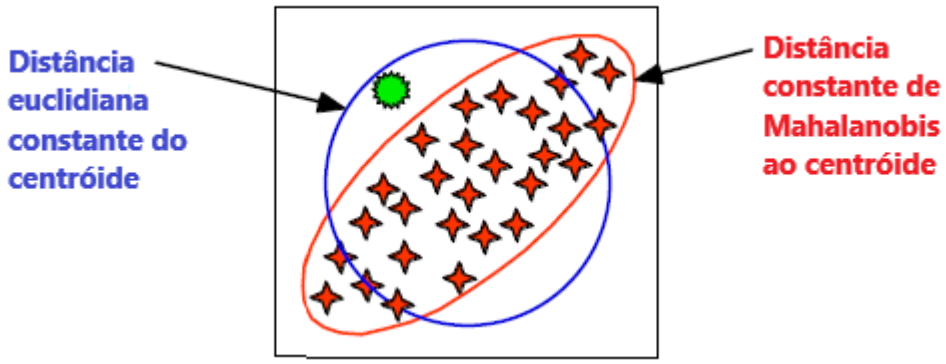


Figura 3. Contraste entre as medidas de distância Mahalanobis e Euclidiana (HOTELLING, 1933).

Para os primeiros k componentes principais, T^2 é definido como:

$$\mathbf{T_i^2} = t_i \mathbf{\Lambda^{-1}} t_i^T = x_i \mathbf{P_k \Lambda^{-1} P_k^T} x_i^T \quad (9)$$

onde $\mathbf{t_i}$ se refere à *i*-ésima linha de $\mathbf{T_k}$, e $\mathbf{\Lambda}$ é a matriz diagonal contendo os correspondentes aos *k* componentes principais descritos no modelo.

Para um espaço de duas dimensões, o limite T^2 define uma elipse dentro do qual os dados normalmente se projetam. Os limites de confiança estatísticos para os valores de T^2 podem ser calculados como:

$$T^2_{k,m,\alpha} = \frac{k(m-1)}{m-k} F_{k,m-k,\alpha} \quad (10)$$

Onde *m* é o número de amostras usadas para desenvolver o modelo PCA, *k* é o número de vetores de componentes principais presentes no modelo e $F_{k,m-k,\alpha}$, é o percentual $(1-\alpha)$ da distribuição de Fisher com *k* e *m* – *k* graus de liberdade. (HOTELLING, 1933)

3.3.3 Bin

O Bin, ou Binning, envolve a integração dos dados espectrais em regiões de igual comprimento, buscando assim minimizar os efeitos das variações nas posições dos picos causadas por efeitos físico-químicos nas amostras. Os compartimentos de dados definidos por cada bin são como a nova matriz de dados utilizada nos cálculos quimiométricos. Aqui é apresentado a comparação entre os diferentes métodos de Bin disponíveis:

3.3.3.1 Método convencional

A extração de sinais espectrais relevantes para análise dos dados leva em consideração os compostos de interesse presentes nos espectros (e.g.: terpenos, glicerol, ácidos graxos, etc.), assim como a utilização de algoritmos de alinhamento de sinais para correção no deslocamento químico dos sinais. Alguns dos métodos de correção de alinhamento comumente utilizados são: Alinhamento por Correlação Otimizada (COW, do inglês Correlation Optimized Warping) (LARSEN; VAN DEN BERG; ENGELSEN, 2006; NIELSEN; CARSTENSEN; SMEDSGAARD, 1998; PRAVDOVA; WALCZAK; MASSART, 2002; SKOV *et al.*, 2006; TOMASI; VAN DEN BERG; ANDERSSON, 2004),Distorção do Tempo Dinâmico (DTW, do inglês Dynamic Time Warping) (PRAVDOVA; WALCZAK; MASSART, 2002; TOMASI; VAN DEN BERG; ANDERSSON, 2004), correlação-deslocamento (coshift) (SAVORANI; TOMASI; ENGELSEN, 2010) e intervalo correlação-deslocamento (icoshift) (SAVORANI; TOMASI; ENGELSEN, 2010; WINNING, 2009). Esses métodos exigem diversos parâmetros para correção dos sinais espectrais, assim como grande conhecimento das distribuições dos sinais presentes nos espectros.

O Bin, ou Bucketing, é mais simples do que os outros métodos citados e mais fácil de ser implementado computacionalmente. No método convencional o espectro é dividido em regiões/*bins* não sobrepostas de tamanho fixo - cuja largura geralmente varia entre 0,01 e 0,05 ppm. Após essa divisão, os valores coletivos de intensidade em cada região/*bins* são substituídos pelo valor integral do bin. (Figura 4). (ANDERSON *et al.*, 2011)

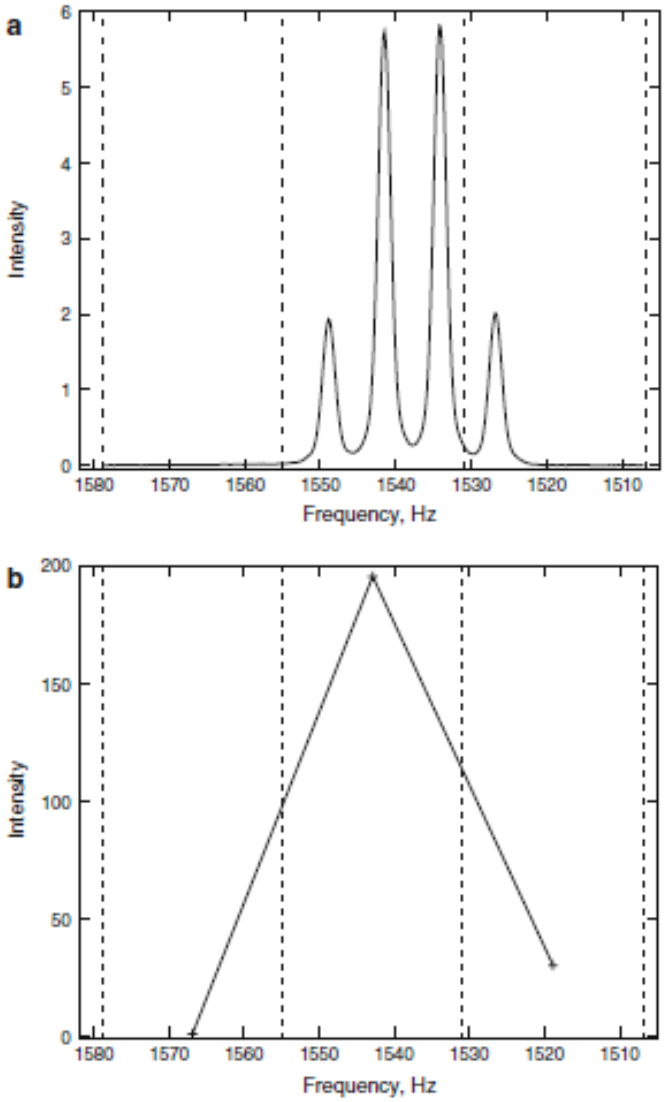


Figura 4. Sinal espectral a) prévio a aplicação do Bin e b) após a aplicação do Bin

Essa técnica busca ao mesmo tempo, reduzir a dimensionalidade dos espectros, i.g. de 32,768 pontos/variáveis para um conjunto com menos variáveis, enquanto tenta minimizar os efeitos das variações nas posições dos picos causadas pelo pH da amostra, força iônica e composição, enquanto reduz a dimensionalidade para análises estatísticas multivariadas (ANDERSON *et al.*, 2011). O produto dessa operação é um conjunto de dados com maior tratabilidade por meio de técnicas de reconhecimento de padrões, como o PCA (BARTHOLOMEW, 2010; HOTELLING, 1933). Contudo, esse procedimento também reduz a resolução dos dados e, se não aplicado corretamente, pode acarretar na perda de informação.

O procedimento de bucketing convencional em uma matriz de dados **X**(I,J) é apresentado na Eq. 11. Cada amostra **i** presente em **X**, x_{ij} (onde i=1, 2, …, **I** e j=1, 2, …, **J**) é um valor de intensidade no sinal bruto no ponto j. O parâmetro **N** representa o número de pontos

de dados em cada bin, sendo calculado pela razão entre a largura do bin e o intervalo de amostragem do espectro. Sousa et al. (2013) traz como exemplo um espectro com intervalo de amostragem de 0,0005 ppm ao qual um bin de 0,05 ppm é aplicado. Nesse caso, o parâmetro N será igual a 0,05/0,0005=100 pontos. Após a integração dos pontos em cada bin, a nova matriz **Z**(I,K). é criada onde novas intensidades $\mathbf{z_{ik}}$ são organizadas. (Equação 11)

$$Z_k = \sum_{j=N*(k-1)+1}^{N*k} x_{ij} \quad k = 1,2,\dots\dots,K \tag{11}$$

3.3.3.2 Algoritmo de Bucketing Otimizado (OBA)

Idealmente, cada bin deve conter um único pico representando o mesmo metabólito. Nos espectros de ^{1}H RMN, um pico representativo de um único tipo de próton pode se dividir em multipletos (ou seja, dupleto, tripleto, etc.) devido ao acoplamento *J*. A desvantagem de se utilizar o bucketing convencional são os limites rígidos delimitados para cada bin. Estes podem dividir um multipleto em dois ou mais *bins* e, consequentemente, separar a informação físico-quimica do sinal em questão em múltiplas variáveis (Figura 4).

Os métodos usualmente empregados para evitar este problema se baseiam em determinar dinamicamente o tamanho e a localização de cada bin de acordo com as dimensões de cada sinal. Uma solução proposta por (DAVIS *et al.*, 2007) , denominada binning adaptativo, utiliza a transformada de onda pequena indecimada (Undecimated Wavelet Transform) para encontrar os mínimos entre sinais em um conjunto de espectros. No Gaussian Binning (ANDERSON *et al.*, 2008), a sobreposição entre os *bins* é controlada pelo desvio padrão do kernel. Já no Dynamic adaptive binning (ANDERSON *et al.*, 2011), os limites de cada bin são calculados a partir da distância média entre os picos, e em seguida, a reposição desses limites é feita considerando a minimização das margens entre dois bin adjacentes.

Apesar de aprimorarem a divisão dos limites de cada bin, esses métodos partilham do mesmo problema: requerem um nível mais alto de experiência do usuário devido à complexidade nas determinações dos parâmetros de input para os algoritmos. (SOUSA; MAGALHÃES; FERREIRA, 2013) O algoritmo de bucketing otimizado (OBA) supera esse problema devido ao baixo número de variáveis para ajuste (largura dos *bins* e *slackness*) por parte dos usuários.

O cálculo dos limites dos *bins* otimizado utiliza o espectro médio transcrito $\mathbf{X}^{\mathrm{T}}$, onde cada elemento x_j é a média da *j*-ésima coluna de **X**(I,J). Dois inputs determinados pelo usuário são levados em consideração no algoritmo: a largura inicial dos *bins*, em ppm, e o quanto o limite pode se mover enquanto busca os mínimos locais no espectro médio (*slackness*), em porcentagem. A largura é transformada em número de intervalos **N**, e o valor do *slackness* é convertido pelo algoritmo no parâmetro *s* por meio da equação s= *slackness* ∗0,01∗N.

Uma vez conhecidos os valores dessas variáveis, cria-se o vetor $\mathbf{v^T}$, cujos elementos definem os limites finais do bin.

$$v^T = [1, \dots, q, \dots, J] \tag{12}$$

Os elementos presentes em **q** no vetor $\mathbf{v^T}$ correspondem ao mínimo local na região delimitada por $\mathbf{x}_{N*t-s}$ e $\mathbf{x}_{N*t+s}$, onde t= 1, 2, …, T, com T sendo igual à parte inteira de (J/N) - 1, como apresentado na equação a seguir. Os elementos de $\mathbf{v^T}$ substituem os limites de integração na Eq. (12), fornecendo assim o bucketing otimizado, para cada amostra i, conforme mostrado na Eq. (14), onde v(k) é o *k*-ésimo elemento do vetor **v**. A nova matriz **Z**(I,K) de intensidades espectrais é obtida para cada amostra presente em **X**.

$$\bar{x}_q = \min\left(\bar{x}_{N*t-s} : \bar{x}_{N*t+s}\right) \tag{13}$$

$$Z_{ik} = \sum_{j=v(k)}^{v(k+1)} x_{ij} \quad k = 1, 2, \dots, k = tamanho\ (v) - 1 \tag{14}$$

Uma questão importante a ser considerada na OBA abordada por Sousa et al, é a escolha da melhor combinação entre a largura dos *bins* e o *slackness* para cada conjunto de dados. A inspeção visual das extensões de desalinhamento na linha de base é fundamental para definição desses parâmetros de forma eficiente. Além disso, alguns critérios, como por exemplo a variância explicada nos primeiros componentes principais de uma análise de componentes principais (PCA), podem ser utilizadas como verificação da divisão mais eficiente dos espectros (SOUSA; MAGALHÃES; FERREIRA, 2013).

4 MATERIAIS E MÉTODOS

4.1 Seleção e preparo de amostras

Para ilustrar a funcionalidade do PCA implementado no GNAT, foi utilizado os dados de RMN de ^{1}H e ^{13}C de amostras de três tipos de óleos vegetais disponíveis comercialmente: Azeite de Oliva, Óleo de Canola e Óleo de Soja. Todos os produtos foram comprados em mercados nas proximidades da UNICAMP. As amostras de azeite tinham como procedência os países: Portugal, Espanha e Argentina. As amostras de óleo de soja e canola possuíam como procedência diferentes estados do Brasil, sendo estes: Goiás, Minas Gerais, Mato Grosso do Sul, Paraná, Santa Catarina e São Paulo. A distribuição do número de amostras para cada tipo de óleo é apresentada na Tabela 2. Ambos os óleos de soja e canola possuíam 9 amostras, enquanto que o azeite possuía 18 amostras, totalizando 36 amostras adquiridas.

Para cada preparação de amostra de RMN, 560 μL de óleo foram misturados em um agitador vórtex, durante 1 minuto, com 140 μL de Benzeno-D6. Em seguida, 600 μL da mistura foram transferidos para um tubo de RMN de 5 mm padrão para medição direta.

Tabela 2. Relação de amostras de óleos vegetais obtidas em estabelecimentos comerciais em Campinas (São Paulo).

Tipo de amostra	Marca	Total de amostras	Origem
Azeite	Oliveira da Serra	18	Portugal
	Oliveira da Serra		Portugal
	Gallo		Portugal
	Gallo		Portugal
	Andorinha		Portugal
	Andorinha		Portugal
	Terrano		Portugal
	Terrano		Portugal
	Herdade do esporão		Portugal
	Herdade do esporão		Portugal
	Carbonell		Espanha
	Carbonell		Espanha
	Pons		Espanha
	Pons		Espanha
	Gomes da Costa		Espanha
	Gomes da Costa		Espanha
	Cocinero		Argentina
	Cocinero		Argentina
Oleo Soja	Liza	9	Minas Gerais
	Liza		Minas Gerais
	Soya		Santa Catarina
	Soya		Santa Catarina
	Vila Velha		Goiás
	Cocamar		Paraná
	Cocamar		Paraná
	Concordia		Mato Grosso do Sul
	Concordia		Mato Grosso do Sul
Oleo Canola	Liza	9	São Paulo
	Liza		São Paulo
	Purilev		São Paulo
	Purilev		São Paulo
	Qualita		Santa Catarina
	Qualita		Santa Catarina
	Soya		Santa Catarina
	Carrefour		São Paulo
	Bunge		Santa Catarina

Como forma de verificar a capacidade de detectar adulterações de óleo de soja nas amostras de azeite e óleo de canola 12 misturas foram preparadas por meio da adição de 25 μL de óleo de soja em 640 μL de marcas diferentes de óleo de canola e azeite do conjunto de 36 amostras. As amostras de canola e azeite selecionadas para adulteração são apresentadas na Tabela 3. No total, 48 amostras foram obtidas e imediatamente analisadas após o preparo.

Tabela 3. Relação de origem das amostras de azeite e canola adulteradas com óleo de soja.

Tipo de amostra	**Origem**	**Total de amostras**	**Marca para adulteração**	**Origem**
Azeite	Portugal Portugal Portugal Espanha Espanha Espanha	6	Soya	Santa Catarina
Oleo Canola	São Paulo São Paulo São Paulo Santa Catarina Santa Catarina Santa Catarina	6	Soya	Santa Catarina

4.2 Obtenção dos espectros de RMN

Os dados foram adquiridos em um espectrômetro Bruker Avance Neo 500 para o experimento de RMN de 1H usando uma sonda de detecção direta (BBO) de 5 mm de diâmetro, a 25 ºC, sem rotação. Os parâmetros de aquisição foram os seguintes: domínio do tempo 32 K, largura de pulso de 90° de 6,5 μs, janela espectral de 13 ppm, tempo de aquisição de 3 s e tempo de relaxamento de 1 s; 32 varreduras e 4 varreduras simuladas foram acumuladas para cada decaimento de indução livre. Os dados para os espectros de ^{13}C RMN foram adquiridos usando largura espectral de 24.039 Hz; 65.536 pontos de dados; largura de pulso de 9,0 μs; atraso de relaxamento de 2,0 s; tempo de aquisição de 1,4 s e 1024 varreduras.

Antes da transformada de Fourier, os decaimentos de indução livres (FIDs) foram preenchidos com zero (*zero-filled*) a 64 k. Um fator de alargamento de linha de 0,3 Hz foi aplicado. Os desvios químicos são expressos em escala δ (ppm), referenciados ao sinal residual do clorofórmio.

4.3 Construção de ferramentas multivariadas

4.3.1 Modulo I – Funções Fundamentais

O Módulo I representa as funções destinadas aos cálculos estatísticos do PCA para determinação das componentes principais. Este módulo tem como principais funções: *pca_svd*, *t2limit* e *qlimit*. Os valores inseridos pelo usuário nas opções de Components e Confidence Value na guia principal do PCA são salvos como as variáveis *ncomp* e *climit*.

A função *pca_svd* é responsável por calcular as componentes principais e valores de T^2 de Hotelling e Q Residual para a matriz de espectros analisada. As funções *t2limit* e *qlimit* utilizam os resultados da função *pca_svd* para cálculo dos limites de T^2 de Hotelling e Q Residual, respectivamente.

Essas funções foram escritas para atuarem independentemente da plataforma gráfica, ou seja, podem ser usadas no *Command Window* no MATLAB pelo usuário somente inserindo os inputs na estrutura adequada para execução de cada função. A estrutura das funções supracitas são apresentadas nos Anexos 1.1,1.2 e 1.3

A rotina escrita na função *BinPCA.m* é baseada no algoritmo desenvolvido por Souza et.al 2013. O código utiliza funções presentes no GNAT para determinar a estrutura da GUI e seu funcionamento, dessa forma, esta função só pode ser executada por meio da estrutura do GNAT. A estrutura desse algoritmo busca apresentar guias visuais para os espectros de RMN carregados no GNAT e para os limites dos *bins* calculados para o espectro médio. A rotina para o cálculo da nova matriz de espectros considerando as variáveis determinadas pelo usuário pode ser visto no fragmento do algoritmo a seguir.

```
function BinPCA = BinPCA(xaxis,X,analyzedata,labels)
       BinPCA = guidata(hBinFigure);
        BinPCA.Bucket=str2double(get(hBucket,'string'));
        BinPCA.Slackness=str2double(get(hSlackness,'string'));

        %Verify user inputs
        if BinPCA.Slackness>100 || BinPCA.Slackness<0
           uiwait(msgbox('The slackness value must be between 0 and
     100%.','Error','error','modal'));
           set(hSlackness,'String',num2str(50));
           return
        end
        %Optimized bucketing calculation
        [p,q]=size(real(X));
        b = xaxis(2)-xaxis(1);
        a = BinPCA.Bucket(1)./b;
```

```
        a = round(a);
        l=BinPCA.Slackness(1)*0.01*a;
        l=round(l);
        R=mean(X);
        v=[];
        for t=1+a:a:q-a
            [~,I]=min(R(t-l:t+l));
            f=((t-(1+a))./a)+1;
            v(f)=I+a*(f-1)+(a-l);
        end
        z = unique(v);
        v = [1 z q];
        Z = zeros(p,(length(v)-1));
        for j = 1:p
            for n = 1:(length(v)-1)
                x_prov = X(j,v(n):v(n+1));
                Z(j,n) = trapzeq(x_prov);
            end
        end
        vv = zeros(p,(length(v)));
        for j = 1:p
            vv(j,:) = X(j,v(:));
        end
        for tt = 2:(length(v)-1)
            Z(:,tt) = Z(:,tt) - vv(:,tt);
        end
        ZNN=Z;
        m=size(Z,1);
        for k=1:m;
            Z(k,:)=Z(k,:)./(sum(Z(k,:)));
        end
        A=[];
        for k=1:length(v);
            A(k)=xaxis(v(k));
        end
        s=length(A);
        I_b=[A(1:(s-1))',A(2:s)'];
        T=[];
        for k=1:(s-1);
            T(k)=A(k)-A(k+1);
            S_b=T';
        end
        function q = trapzeq(y)
            n = length(y);
            sums = y(1) + 2*sum(y(2:n-1)) + y(n);
            q = sums/2;
        end
variables=[1:size(ZNN,2)];
BinPCA.number_of_variables = variables;
BinPCA.intervals = I_b;
BinPCA.non_normalized_buckets = ZNN;
BinPCA.normalized_buckets = Z;
BinPCA.size = S_b;
line([I_b(:,1)              I_b(:,1)],[BinPCA.Ylimits(1)              %Lines
BinPCA.Ylimits(2)],'color',[0 0 0 0.25],'LineWidth', 0.9,'tag','I_b');
```

A proposta das funções presentes na GUI *BinPCA.m* é gerar uma matriz de espectros corrigidos para serem usados na função *pca_svd*. Assim, o algoritmo só permite ao usuário

prosseguir com a análise se os parâmetros de *Bucket e Slackness* estiveram preenchidos em uma estrutura adequada para execução das funções. Caso os parâmetros não sejam adequados para esse cálculo, a mensagem de erro presente na figura 5 aparece como um alerta.

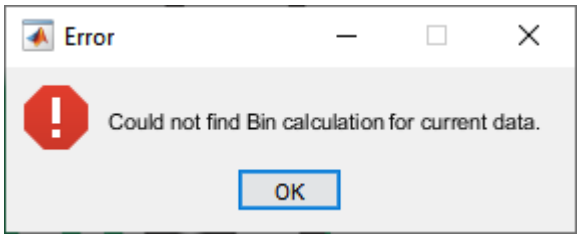


Figura 5. Mensagem de erro padrão para inputs incongruentes na aplicação dos limites Bins.

Caso o usuário deseje desfazer o pré-processamento aplicado aos dados espectrais, é necessário somente selecionar o botão X na GUI aberta (Figura 6). Dessa forma, a matriz com os limites dos *bins* é excluída e a matriz original é resgata na estrutura do GNAT para repor esses dados.

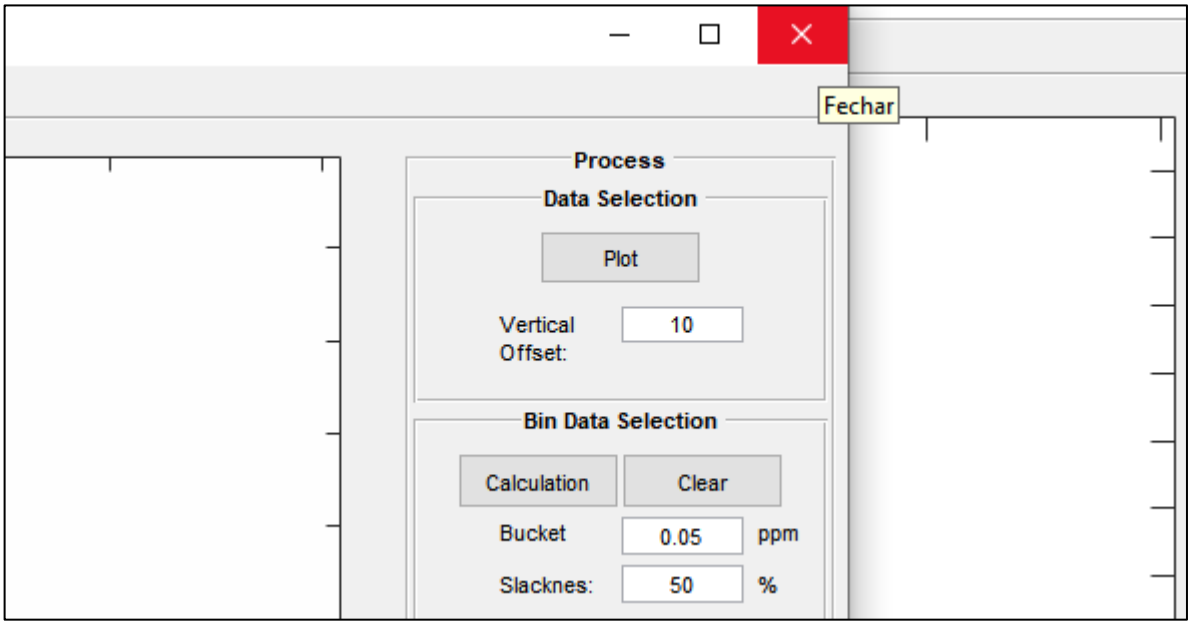


Figura 6. Procedimento para remoção dos pré-processamentos selecionados do Bin para a matriz de dados.

Um conjunto de funções que permitem a melhor visualização dos resultados estão disponíveis na construção de classes por meio da guia *Class (Optional)*. Como se trata de uma ferramenta opcional, existem valores padrões para classes do conjunto total de amostras caso nenhum input do usuário seja fornecido – todas serão definidas como classe 1. A Figura 7 demonstra as opções disponíveis para criação dos vetores de classe.

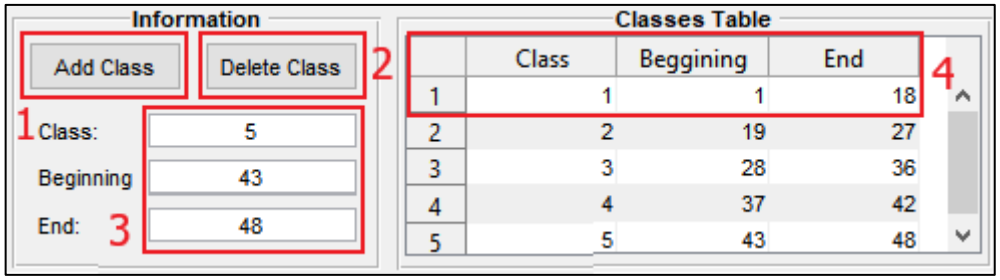


Figura 7. Construção da matriz de classes para os espectros carregadas no GNAT.

A adição das classes é realizada pelo *pushbutton* em '1'. Esse botão irá analisar os valores inseridos em '3' para adicionar a classe na *Classes Table*, em '4'. O código foi escrito considerando todas as opções de possíveis inputs fornecidas pelo usuário para um correto funcionamento do programa. A figura 8 apresenta o exemplo de algumas mensagens de aviso que aparecem para o usuário caso alguma das três opções de entradas em '3' possuam um valor incongruente com o conjunto de dados.

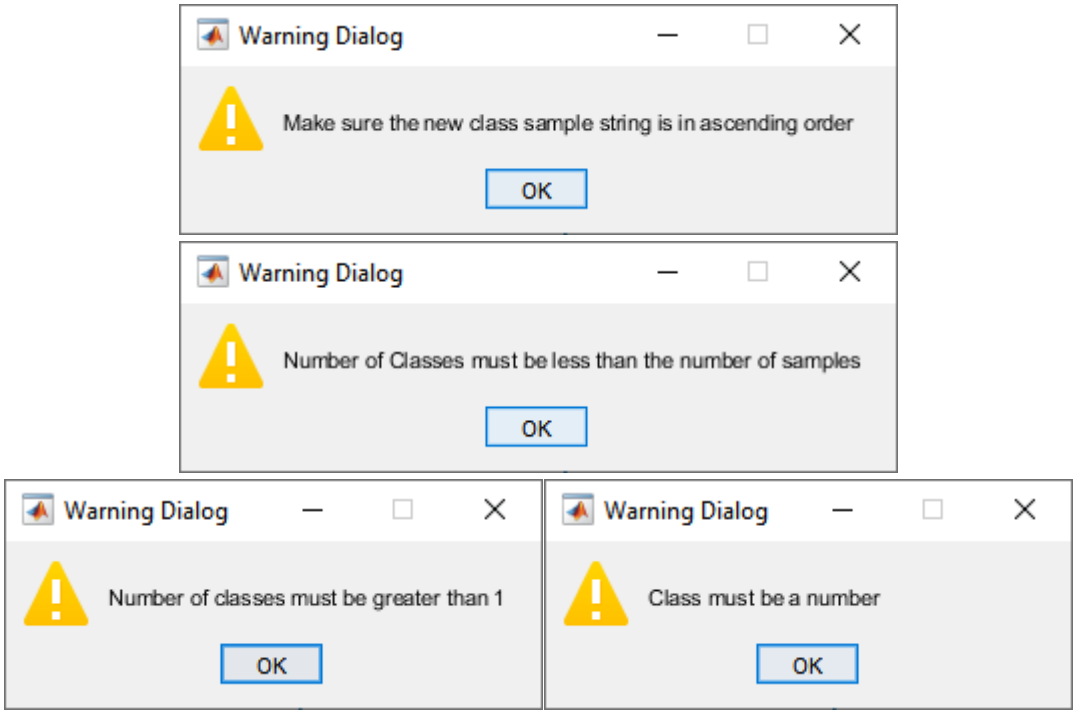


Figura 8. Mensagens de aviso após de detecção de erro nos inputs para construção de Classes.

Para deletar uma classe criada basta adicionar o número dessa classe na *edit* box *Class* em '3' e então pressionar o botão *Delete Class* '2', esse procedimento funciona independentemente da posição da classe. Para deletar a última classe criada na *Classes Table* é necessário deixar a *edit box Class* vazia e, em seguida pressionar, o botão *Delete Class*.

As seguintes regras são definidas na construção das classes:

- A amostra que inicia a classe (Beggining) deve vir previamente da amostra que determina o final da classe (End);
- O número máximo de classes possíveis de serem construídas é limitado pelo número de amostras carregadas no GNAT;
- O identificador da classe deve ser representado por um número, onde a numeração que define a classe deve ser igual ou maior que um.

As funções dentro da aba *Class(Optional)* foram escritas para permitir que amostras dispersas no conjunto de dados possam estar em uma mesma classe, lidando dessa forma com a possibilidade dos conjuntos de espectros não estarem organizados em ordem decrescente. Os controles que permitem essas operações aparecem na figura 9.

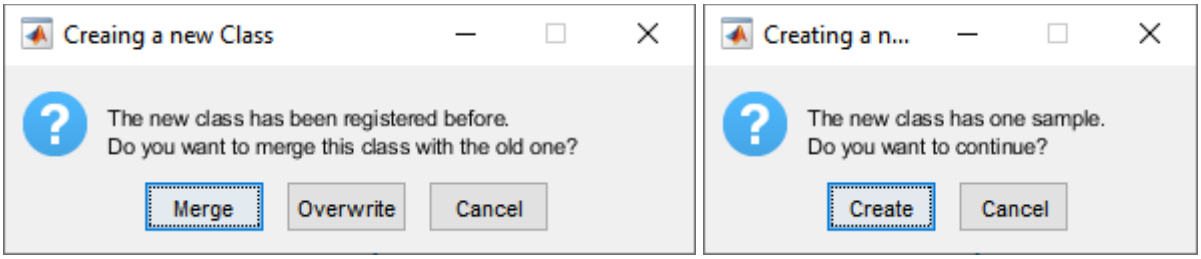


Figura 9. Mensagens de aviso após a seleção do botão Add Class para casos onde a detecção de valores de construção de classes descreve amostras dispersas.

4.3.2 Modulo II – Visualização dos Resultados

O Módulo II representa o conjunto de funções nessa nova GUI, utilizadas na visualização dos gráficos de Escores, Pesos e Resíduos calculados e salvos no modelo PCA. É possível dividir esse conjunto de funções entre as responsáveis pela estrutura da GUI e as responsáveis por construir o conteúdo dos gráficos. O conjunto de funções para essa construção são: *ScoreCheck_Callback*, *EnableZaxis_Callback*, *ChangingScore_Callback* e *EnableLabel_Callback*. Estas funções são executadas após a interação do usuário com os objetos gráficos da GUI e são descritas como os *callbacks* dos objetos gráficos.

Como exemplo, tem-se o fragmento da rotina da estrutura *Labels*, onde a função *EnableLabel_Callback* é executada após a seleção do objeto gráfico de estilo *Checkbox*, na interface gráfica

```
hCheckLabels = uicontrol(...
'Parent',OptionsPlotPanel,...
'Style','Checkbox',...
'Units','normalized',...
'horizontalalignment','l',...
```

```
'TooltipString',...
'Labels for each point',...
'Value',1,...
'Position',[0.05 0.4 0.1 0.4],...
'BackgroundColor',[1 1 1],...
'Callback',{@EnableLabel_Callback}); % Enable/Disable Label visualization
```

Um callback é uma função executada em resposta a alguma ação predefinida do usuário, como clicar em um objeto gráfico ou fechar uma janela de figura. Todos os objetos gráficos possuem propriedades para as quais é possível definir os seguintes tipos de callback:

- *ButtonDownFcn* – Executado quando o usuário pressiona o botão esquerdo do mouse sobre um objeto gráfico interativo ('checkbox', 'radiobutton', 'edit' e 'popupmenu');
- *CreateFcn* - Executado durante a criação de um novo objeto após o MATLAB® definir todas as propriedades deste;
- *DeleteFcn* - Executado logo antes do MATLAB excluir o objeto.

As funções *EnableZaxis_Callback, ChangingScore_Callback e EnableLabel_Callback* são responsáveis, respectivamente, por: transformar o plot dos Escores em 2D para um plot em 3D; ler os valores selecionados pelo usuário para as PCs que serão apresentadas no eixo x,y e z e controlar a visualização da numeração das amostras presentes no conjunto de dados plotado.

A função *ScoreCheck_Callback* é a responsável por gerenciar as opções selecionadas pelo usuário para construção das diversas opções de construção gráfica disponíveis na GUI.. Os exemplos de plots possíveis para visualização dos dados são apresentados na Figura 10. A estrutura resumida das funções do Modulo II é apresentada nos Anexos 2.1, 2.2, 2.3 e 2,4

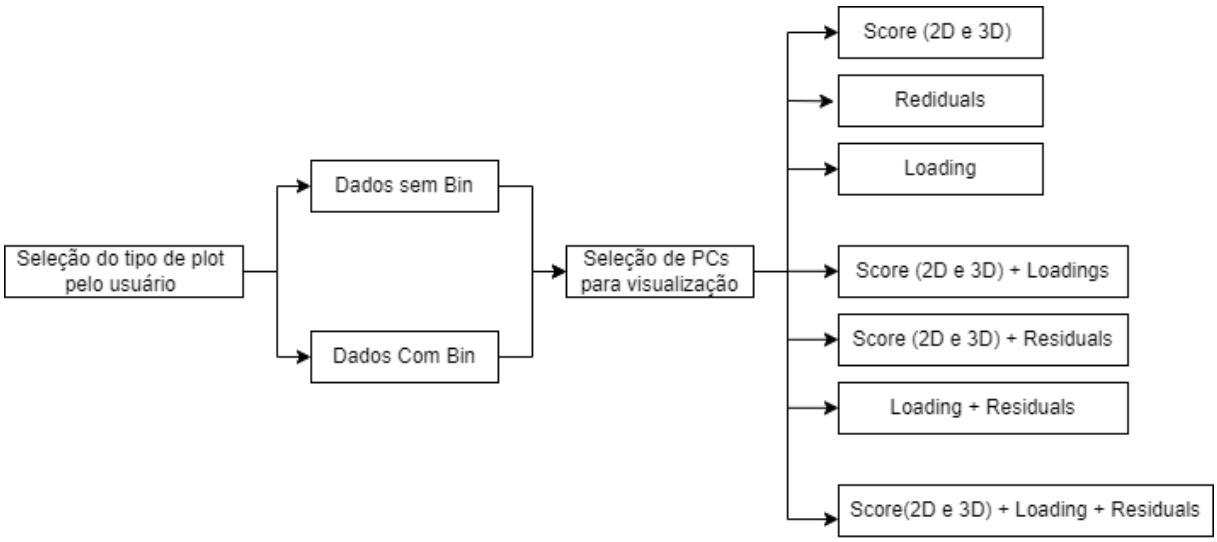


Figura 10. Fluxograma dos diferentes tipos de modos de visualização dos resultados.

4.4 Validação do Software

A validação das funções de pré-processamento de RMN e cálculos quimiométricos escritas para a estrutura do GNAT foi realizada comparando os resultados de componentes principais, plots de escores e plot de pesos processados no GNAT com os espectros de RMN processados pelo software PLS_toolbox (versão 8.8.1, Eigenvector Inc.) (EIGENVECTOR, 2022)

5 RESULTADOSE DISCUSSÕES

5.1 Analise dos óleos vegetais

5.1.1 RMN de ^{1}H e ^{13}C

As Figuras 11 e 12 mostram o conjunto de espectros de RMN de 1H e 13C dos óleos vegetais analisados. Os sinais dos compostos majoritários nos espectros de RMN são atribuídos nas Tabela 4 e 5 (Sacchi et al., 1997; Vigli et al., 2003, propescu 2020). Devido à semelhança química dos diferentes ácidos graxos presentes nessas amostras, a maioria sinais dos presentes no RMN de ^{1}H se sobrepõem e não permitem a atribuição inequívoca dos sinais de cada ácido graxo. (por exemplo, os sinais em: 0,87, 1,30 e 1,62). Em contrapartida, os sinais em 14,0, 20,0 e 25,0 ppm no ^{13}C RMN, referentes aos grupos metil, metileno e etileno, são bem espaçados e permitem a melhor identificação de seus deslocamentos.

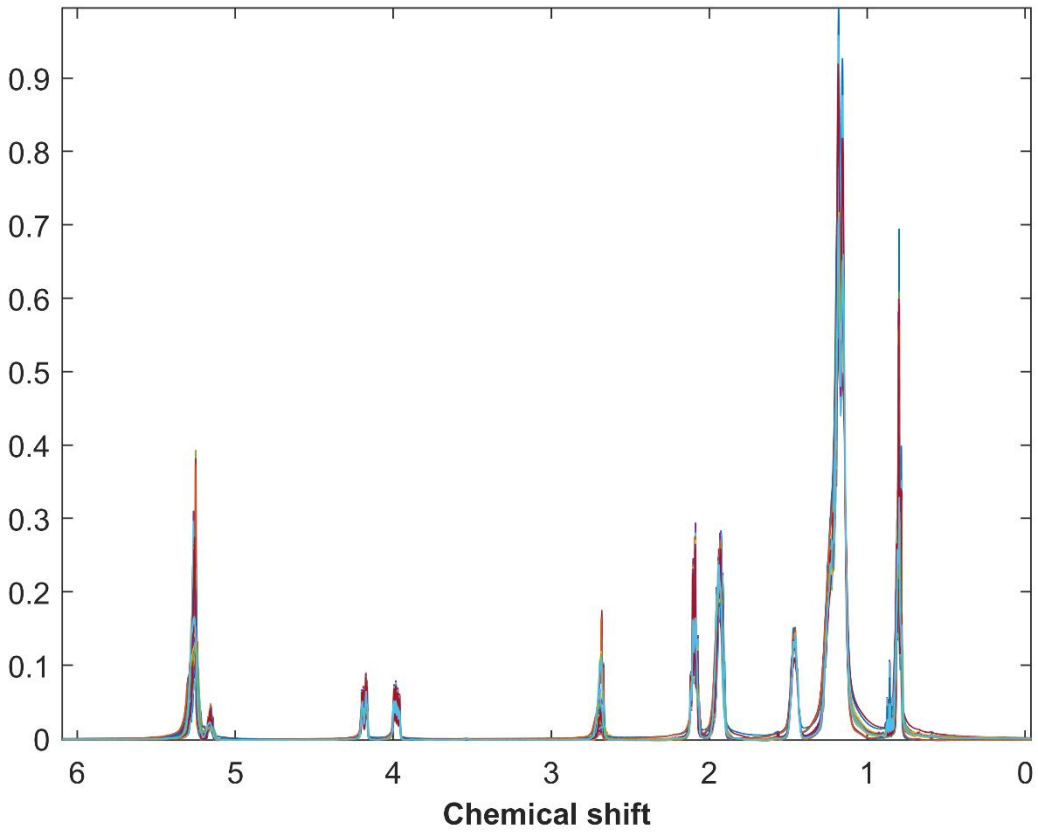


Figura 11. Sobreposição dos espectros de RMN de ^{1}H (600 MHz) das amostras de óleo de oliva, soja e canola.

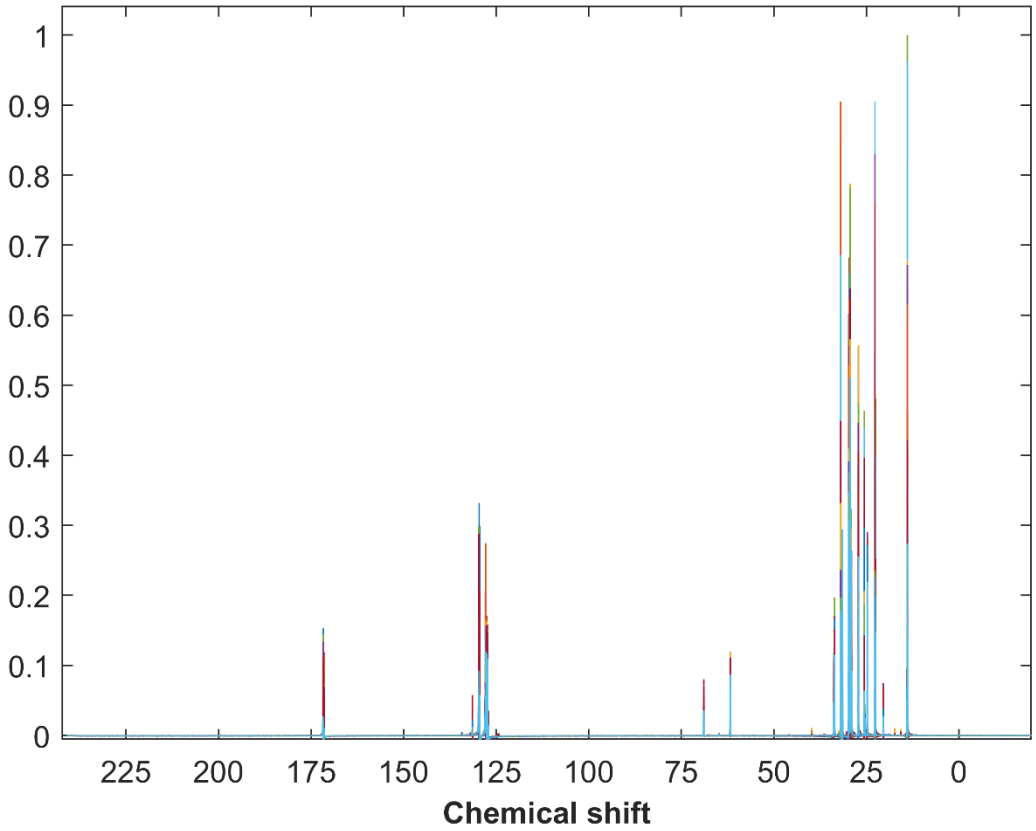


Figura 12. Espectros de RMN de ^{13}C (600 MHz) das amostras de óleo de oliva, soja e canola.

As Tabela 4 e 5 reúne os sinais de maior intensidade dos espectros de RMN de ^{1}H e ^{13}C para o conjunto de óleos analisados, juntamente com seus deslocamentos químicos e suas atribuições aos átomos dos diferentes grupos funcionais, como já observado por outros autores (Alonso-Salces, Heberger et al., 2010; Alonso-Salces, Moreno-Rojas et al., 2010; D'Imperio et al., 2007; Guillen & Ruiz, 2001; Mannina, Sobolev, & Segre, 2003; Sacchi et al., 1996)

Tabela 4. Relação dos sinais presentes nos espectros de RMN de ^{1}H com seu deslocamento químico e composto associado.

Deslocamento químico (ppm)	^{1}H	Composto
0,83-0,93	-C**H**3	Hidrogênio terminal (Oleico e Linoleico)
0,93-1,03	-C**H**3	Hidrogênio terminal (Linolênico)
1,14-1,42	-(C**H**2)n-	Grupos Acila
1,54-1,68	-OCO-CH2-C**H**2-	Grupos Acila
1,96-2,08	-C**H**2-CH=CH-	Metilenos Alila
2,26-2,38	-OCO-C**H**2-	Metilenos Carboxílicos
2,74-2,83	=**H**C-CH2-C**H**=	Metilenos Dialílicos
3,70-3,75	C**H**2O-	Diacil glicerol-1,2
4,10-4,38	-C**H**2OCOR-	Glicerol
5,24-5,29	> C**H**OCOR	Glicerol
5,30-5,41	-C**H**=C**H**-	Hidrogênios Vinílicos

Como descrito na literatura por diversos autores, vários sinais de compostos menores presentes nos espectros de 1H-RMN registrados para os diferentes tipos de óleos foram sobrepostos pelos sinais de prótons de compostos majoritários, como por exemplo: trigliceril: cicloartenol a 0,32 ppm e 0,54 ppm, b-sitosterol a 0,67 ppm, estigmasterol a 0,69 ppm, esqualeno a 1,66 ppm, prótons do grupo sn-1,2 digliceril a 3,71 ppm e 5,10 ppm e três terpenos desconhecidos a 4,57 ppm, 4,65 ppm e 4,70 ppm. (ALONSO-SALCES; HOLLAND; GUILLOU, 2011). Os sinais de ressonância dos espectros de RMN de 13C dos óleos podem ser agrupadas em quatro regiões espectrais: carbonos carbonílicos (173,3 a 172,8 ppm); carbonos insaturados (132,1 a 126,8 ppm); carbonos de glicerol (69,1 a 61,6 ppm); e carbonos alifáticos (34,5 a 13,9 ppm). (DI PIETRO; MANNU; MELE, 2020; SHAW *et al.*, 1997)

A numeração dos átomos de carbono nos ácidos graxos se inicia no carbono carbonilico e finaliza no carbono metil final na cadeia conhecido como omega (ω) ou n. Dessa forma, o ácido linoléico é um ácido graxo ω-6 (n-6) devido sua última ligação dupla ter 6 carbonos de distância até a extremidade metil da molécula (Figura 13). Para indicar as posições das ligações

duplas é usado o símbolo delta (Δ) seguido pelo número sobrescrito. Por exemplo, Δ^9 significa que existe uma ligação dupla entre C-8 e C-11

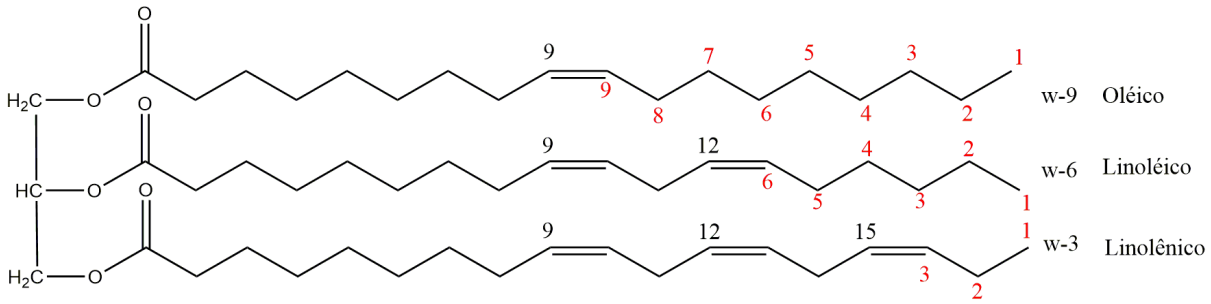


Figura 13. Estrutura dos ácidos graxos presentes em um triacilglicerol (TAG) com indicações de saturações na molécula e identificação da cadeia carbônica pelo símbolo ômega (ω)

Tabela 5. Relação dos sinais presentes nos espectros de RMN de ^{13}C com seu deslocamento químico e composto associado.

Deslocamento químico (ppm)	^{13}C	Composto
14,1 – 14,3	C18(ω1)	Todas as cadeias Acila
20,6 – 22,7	C17(ω2)	Todas as cadeias Acila
24,5 – 24,9	C3	Todas as cadeias Acila
25,6 – 25,7	C11	Linoleico e Linolênico
	C14	Linolênico
27,2 – 27,4	C8	Oleico e Linoleico
	C11	Oleico
29,4	C4 - C7	Todas as cadeias Acila
	C12 – C15	Oleico
	C8 – C15	Stearoil
31,6 – 31,9	C16(ω3)	Linoleico
34,1 – 34,2	C2, sn-2	Todas as cadeias Acila
62,1	CH2O–, sn-1,3	Glicerol (triacilgliceróis)
	CH2O–, sn-1	Glicerol (1,2-diacilgliceróis)
65,1	CH2O–, sn-1	Glicerol (Monoacilgliceróis)
68,9	CHOe,	sn-2 Glicerol (Triacilgliceróis)
77,0	CDCl3	Solvente
127,1 – 127,9	C12	Linoleico
	C10, C15	Linolênico
128,1 – 128,5	C10	Linoleico
	C12, C13	Linolênico
129,5 – 129,7	C9, C10	Oleico
130,0 – 130,5	C9	Linoleico e Linolênico
	C13	Linoleico
171,4	C1, sn-2	Triacilgliceróis
171,7	C1, sn-1,3	Triacilgliceróis

5.2 PCA Painel

A Figura 13 mostra a aba principal da GUI para PCA que foi implementada no ambiente GNAT. Os painéis e abas disponíveis no PCA fornecem os controles de usuário para permitir a seleção dos parâmetros de pré-processamento para cálculo (Bin, Componentes e Valor de

Confiança) e dos aspectos da exibição dos resultados (*Plots panel*, *Variance Captured by PCA*, e *Classe (Opcional)*). As funções utilizadas previamente a aplicação do PCA presente no GNAT abrangem as seguintes categorias gerais: carregamento de dados (Bruker, Varian, Jeol Generic e Spinach) e pré-processamento. As funções de pré-processamento construídas no início da formulação do GNAT incluem métodos para Transformada de Fourier, Preenchimento zero, Apodização, Correção de fase (CRAIG; MARSHALL, 1988) e Correção da linha de base (PEARSON, 1977). Mais informações sobre essas operações podem ser obtidas em (CASTAÑAR *et al.*, 2018).

Os painéis contendo os controles dos Componentes e o Valor de Confiança são editáveis, e representam o número de PCs disponíveis para exibição nos plots de scores e loadings e o percentual de confiança utilizado no cálculo dos limites de T^2 e Q Residuals, respectivamente. (Figura 13). Já a tabela de Variância Capturada pelo PCA contém os resultados do PCA expressos pela variância explicada e a variância acumulada apresentada em cada PC (componentes principais). As PCs são ordenados na tabela em ordem decrescente de sua variância total explicada.

Escores e pesos podem ser analisados em uma GUI própria, onde diferentes opções são dadas ao usuário para explorar os resultados. O painel *Plots* seleciona qual resultado será mostrado nessa GUI para visualização dos resultados – *Scores*, *Loadings* e *Residuals*.

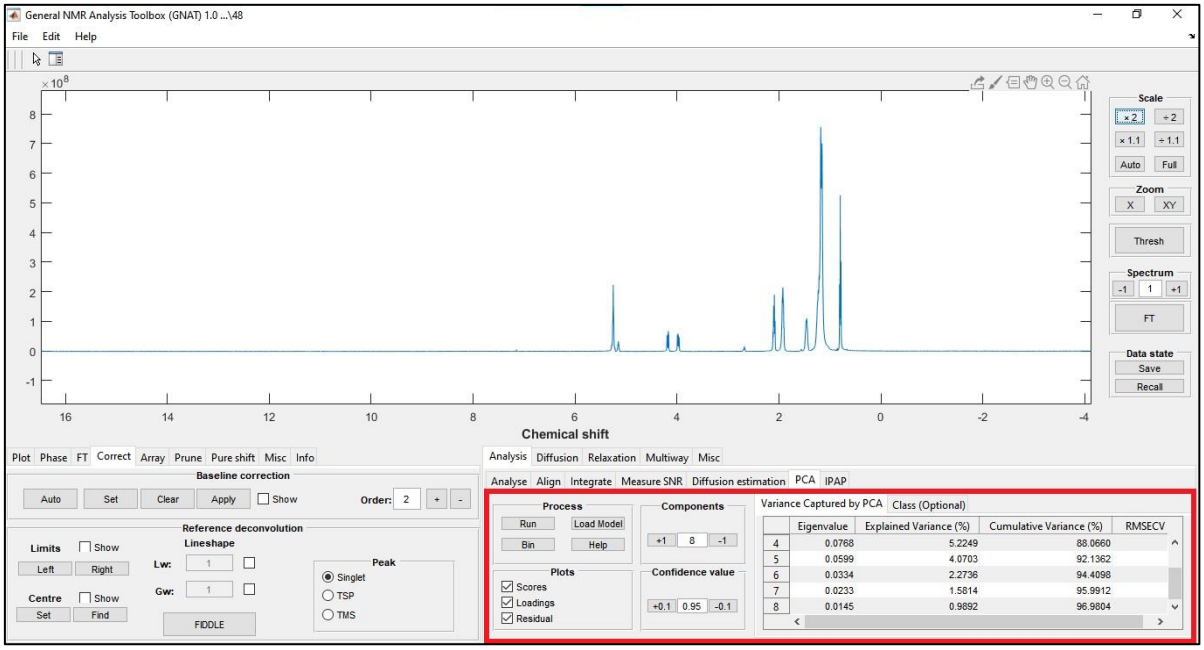


Figura 14. Na análise dos dados (grupo da aba direita) é mostrada a aba PCA, a nova ferramenta quimiométrica implementada no GNAT. O espectro mostrado é um espectro de 500 MHz ^{1}H RMN de azeite de oliva.

5.2.1 Class painel

A guia *Class (Optional)* inclui ferramentas para permitir a inclusão da informação de classes presentes na relação de amostras carregadas no GNAT. Como apresentado anteriormente, não há a necessidade de construir classes para cada análise, tendo em vista que o PCA se trata de um modelo não supervisionado (os modelos encontram os padrões presentes nos dados fornecidos sem a necessidade de informações externas fornecidas pelo usuário). Contudo, essa etapa permite uma melhor visualização da separação das amostras nos gráficos de escores e residuais – como parâmetro padrão, todas as amostras são definidas como classe 1.

Para o conjunto de dados analisado, foram construídas 5 classes: Classe 1: Azeite; Classe 2: Óleo de Soja; Classe 3: Óleo de Canola; Classe 4: Azeite adulterado com Óleo de Soja e Classe 5: Óleo de Canola adulterado com Óleo de Soja (Figura 14).

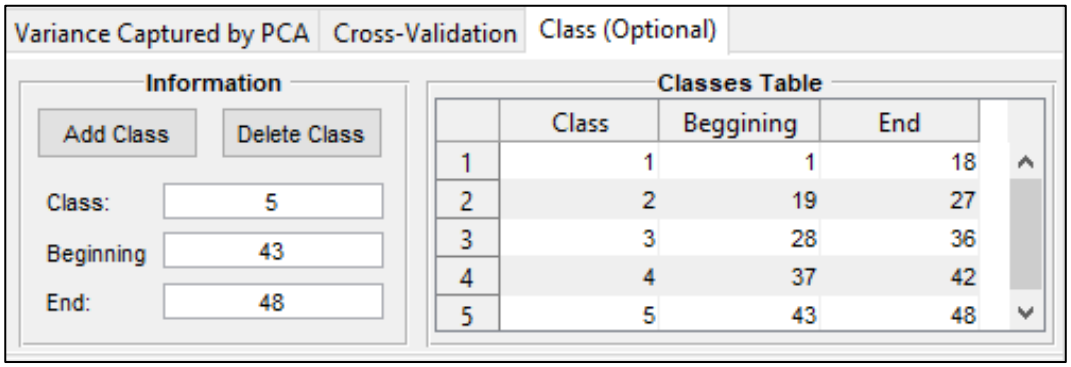


Figura 15. Aba de construção de classe. O número de classes é limitado ao número de amostras; no entanto, não é necessário construir classes em todas as análises.

5.3 Binning GUI

No ambiente GNAT, existem diversas ferramentas de pré-processamento que resolvem problemas como distorção de fase, alinhamento de pico e correção de linha de base. (CASTAÑAR *et al.*, 2018). Para análise de PCA, a maioria dos trabalhos utilizam o bin como um método de seleção de variáveis, resultando em um conjunto de dados mais suscetível à tratabilidade de técnicas de reconhecimento de padrões (ANDERSON *et al.*, 2011; BARTHOLOMEW, 2010; HOTELLING, 1933).

A figura 15 mostra a implementação do algoritmo de binning otimizado (OBA) como descrito por Sousa et al, na estrutura do GNAT. Os painéis contendo os controles dos *Plots*, *Calculation* e *Clear* são editáveis, e permitem, respectivamente: visualizar os espectros carregados no GNAT; calcular os limites dos *bins* de acordo com os valores de *bucket* e *slackness*; e limpar o último cálculo dos limites dos *bins* determinado pelo usuário. A utilização do bin convencional é feita selecionando o valor de *slackness* como 0%.

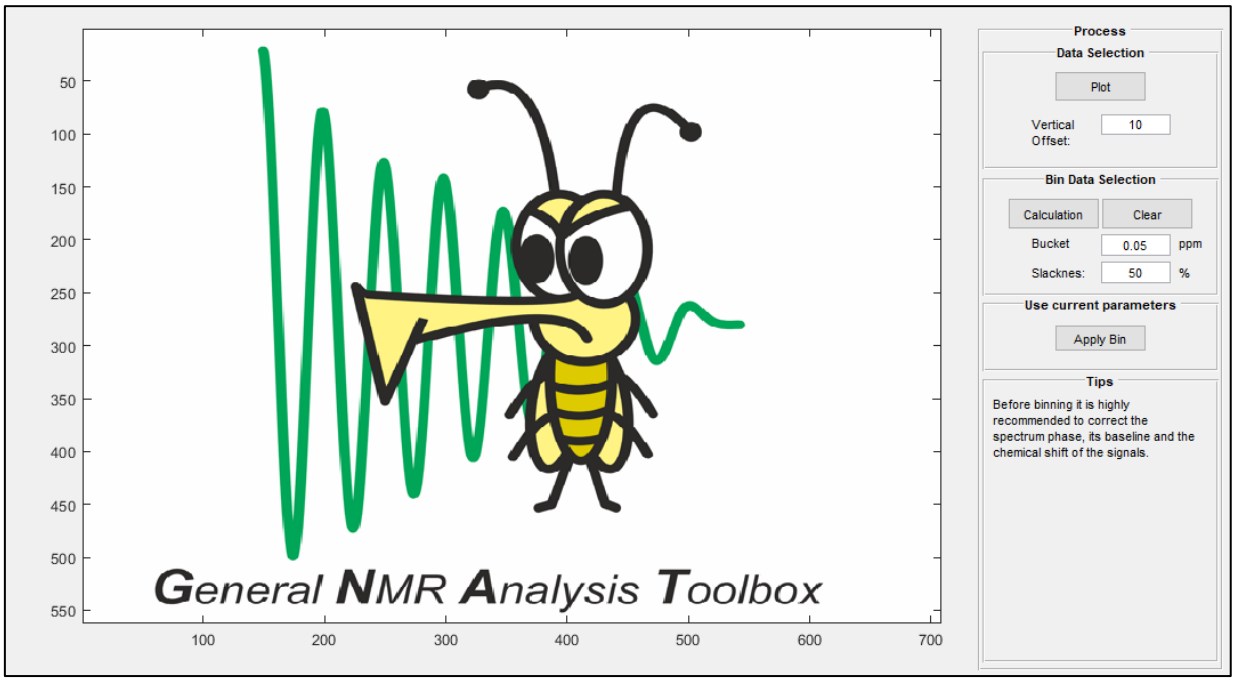


Figura 16. O ambiente de binning para realiza uma redução de dados agrupando respostas espectrais em bins individuais. O código implementado é baseado no algoritmo de agrupamento otimizado (OBA).

A figura 16 ilustra a comparação entre o bin convencional e algoritmo de binning otimizado para o conjunto de dados de óleo vegetais. A largura do bin para um valor de 0,05 ppm foi selecionada devido à melhor divisão dos limites entre os sinais nos espectros. Valores maiores que 0,05 ppm para os limites do bin auxiliariam na correção de desalinhamento de sinais maiores; no entanto, essa abordagem leva a uma inerente diminuição da resolução no eixo de deslocamento químico, portanto, na seleção desses limites deve haver um compromisso entre o ganho de correção de desalinhamentos e a redução do número de variáveis. Para o conjunto de dados utilizados neste trabalho, os parâmetros de entrada foram determinados inspecionando os desalinhamentos da linha de base e os gráficos dos bin obtidos, a fim de escolher aqueles com menor decréscimo na resolução.(SOUSA; MAGALHÃES; FERREIRA, 2013).

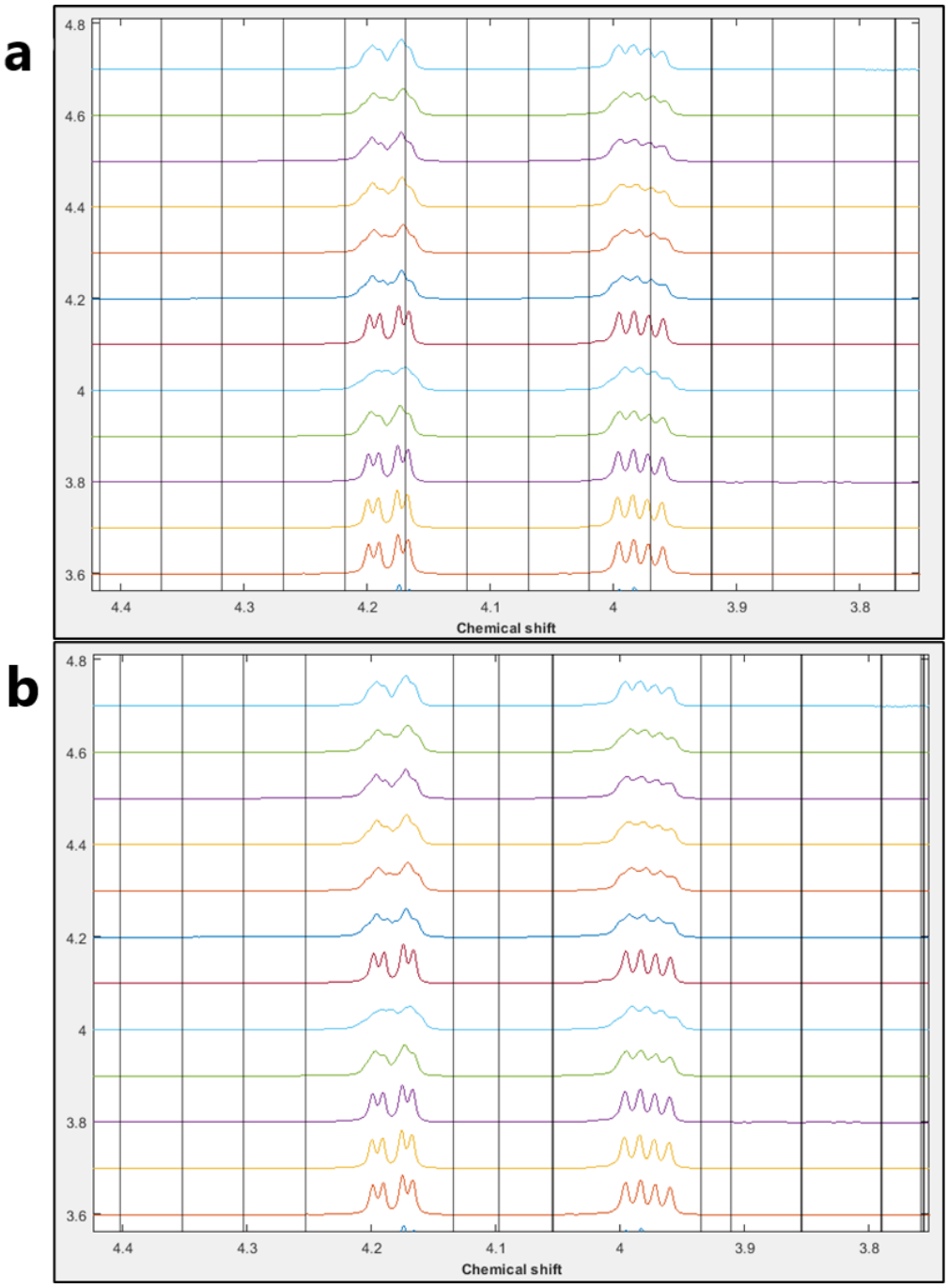


Figura 17. (a) Bucking convencional e (b) Limites de bucketing otimizados de espectros de RMN de ^{1}H de deslocamento puro de amostras de óleo processadas com o software GNAT. Os parâmetros para os compartimentos individuais do algoritmo de agrupamento otimizado foram a largura inicial do balde em ppm: 0,05 ppm e o *slackness* (a porcentagem de quão longe o limite pode se mover enquanto procura os mínimos locais) de 70%.

Para o conjunto teste de óleos vegetais, os espectros de ^{1}H RMN organizados em uma matriz de dados possuíam as dimensões de 48×32.768. A largura espectral utilizada no cálculo do bin compreendia a região de 0,02 a 10,00 ppm, sendo reduzida em *bins* com largura de 0,05 ppm e 50% de *slackness*. Os espectros de RMN de 13C foram dispostos em uma matriz com as mesmas dimensões. Contudo, a largura dos espectros utilizada no cálculo do bin compreende a

região de 0,02 a 185,00 ppm, sendo essa reduzida em *bins* com largura de 0,5 ppm e *slackness* de 70%. Dessa forma, as matrizes dos espectros de ^{1}H e ^{13}C foram reduzidas a matrizes de (48×399) e (48×458) variáveis, respectivamente.

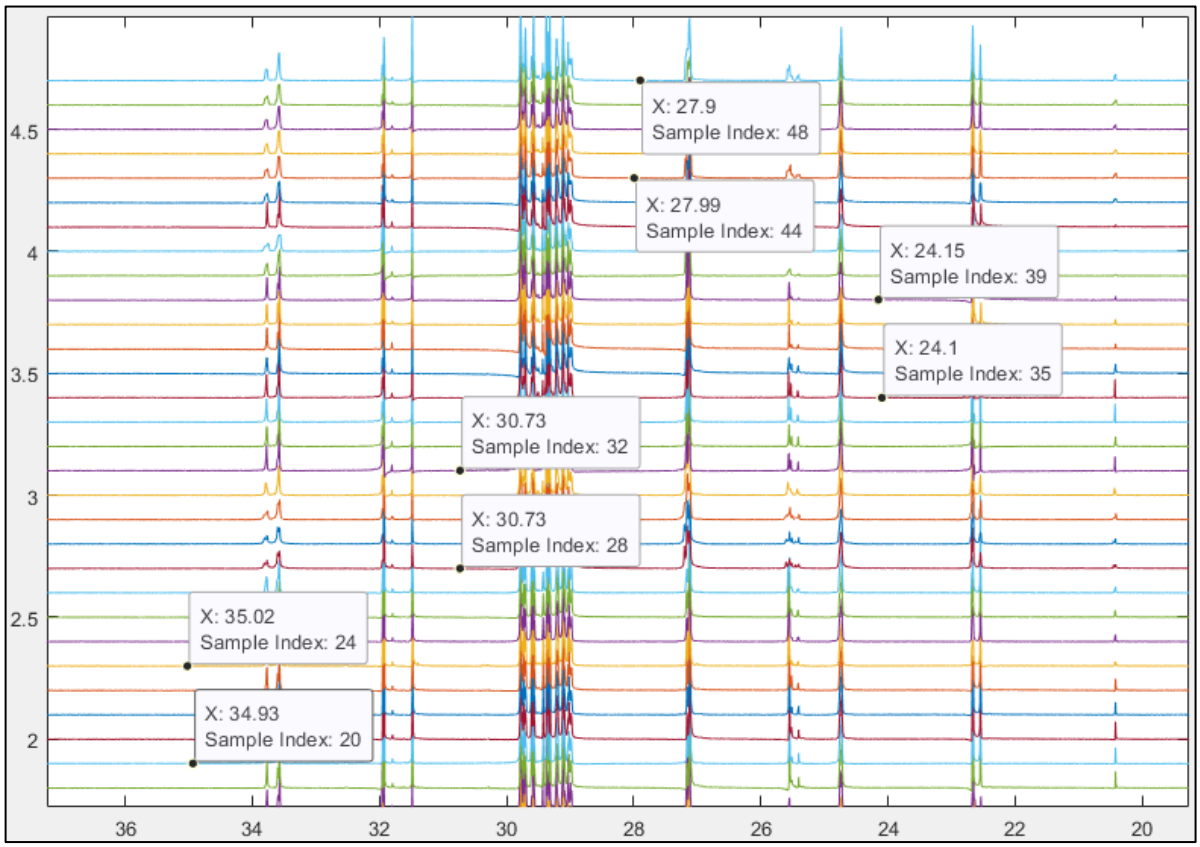


Figura 18. Apresentação da caixa de texto guia para auxiliar na identificação do número do espectro e deslocamento químico espectral para os espectros de óleos vegetais de ^{13}C.

Um guia visual desenvolvido para identificação dos espectros distribuídos na GUI Bin é apresentado na Figura 17. A caixa de texto guia aparece em cada espectro que o usuário seleciona, assim como e deslocamento químico espectral referente a posição no eixo X selecionado. A seleção de múltiplas caixas de texto é possível segurando a tecla *shift*. O algoritmo para construção dessa caixa de texto é apresentado a seguir:

```
function output_txt = myupdatefcn(obj,event_obj)
% Display the position of the data cursor
% obj          Currently not used (empty)
% event_obj    Handle to event object
% output_txt   Data cursor text string (string or cell array of strings).

  old_data=guidata(obj);
  pos = get(event_obj,'Position');
  [row,col]=find(old_data.SPECTRA==(pos(2)));
  output_txt = {['X: ',num2str(pos(1),4)],['Sample Index: ',num2str(col)]};
end
```

Uma comparação entre os resultados do PCA para os dados de ^{1}H com e sem binning é apresentado na Figura 18. Nela é possível ver que a separação dos conjuntos de amostras de azeites e óleos por meio da PC1 é menor para o conjunto de amostras da matriz 48×32.768. Além disso, a variância explicada no conjunto de espectros após o binning para a PC1 e PC2 é maior em relação ao conjunto sem binning.

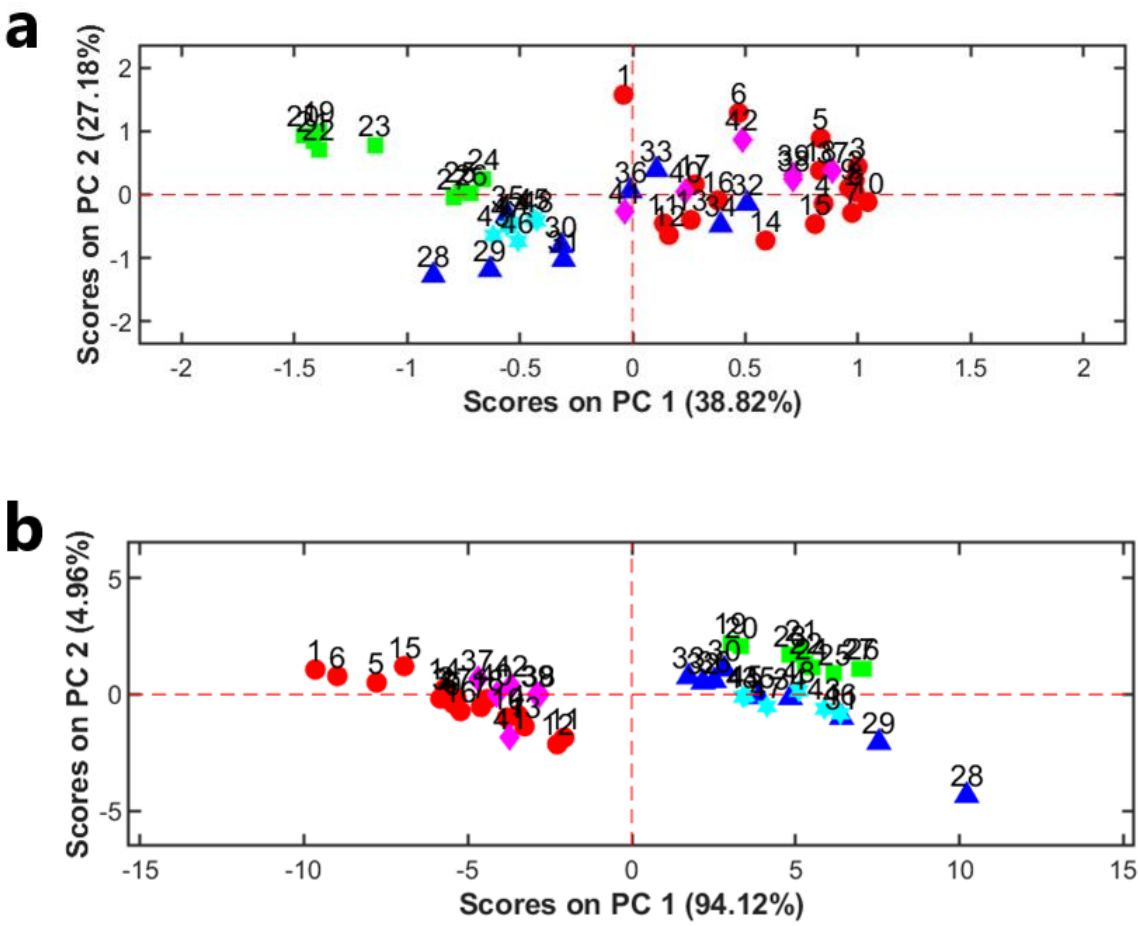


Figura 19. Comparação entre a PC 1 vs PC 2 para os dados de a) ^{1}H-RMN sem bin b) com bin de 0,5 ppm e 50% de *slackness*. (●) Azeite (♦) Azeite adulterado com óleo de soja, (▲) Óleo de canola, (✶) Óleo de canola adulterado com óleo de soja e (■) Óleo de soja.

5.4 PCA Plots GUI

Os resultados do PCA podem ser visualizados na interface gráfica presente na Figura 19. Um grupo de botões na parte esquerda do painel fornece aos usuários o controle dos PCs exibidos nos gráficos de escores, pesos e residuais na parte central da GUI. O gráfico de escores

representa as coordenadas das amostras no espaço das PCs, permitindo a investigação visual da estrutura de dados analisando as posições da amostra e seus relacionamentos.

As amostras carregadas com uma classe definida apresentarão a legenda do seu grupo quando a caixa C*lass* estiver selecionada, o mesmo para a numeração das amostras carregadas no GNAT por meio da caixa *Labels*. Já nos plots dos pesos (*Loadings*) é apresentado os sinais espectrais (variáveis) responsáveis pelos agrupamentos nos plots dos escores.

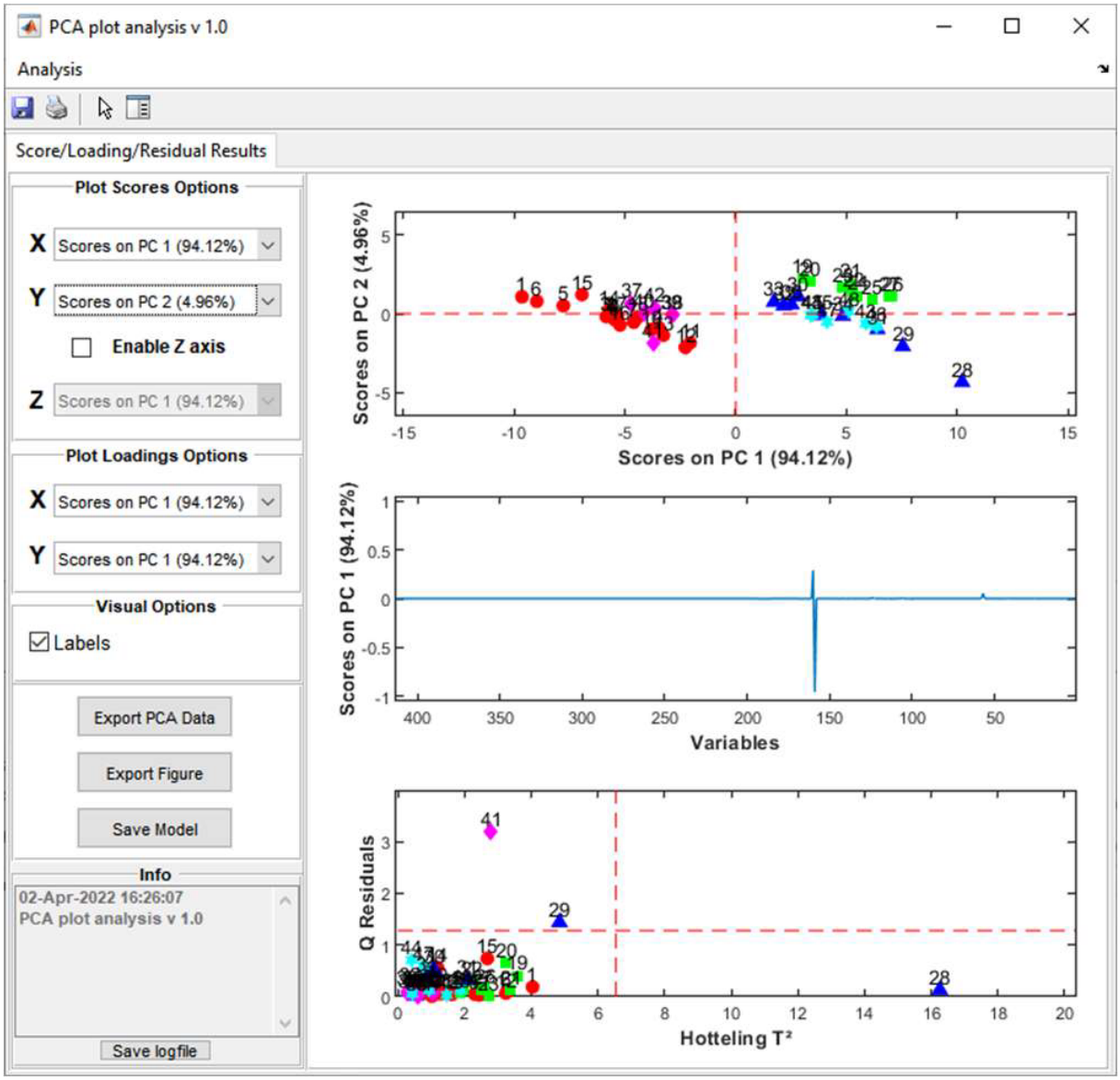


Figura 20. Estrutura da GUI para visualização dos resultados dos cálculos PCA. (●) Azeite (♦) Azeite adulterado com óleo de soja, (▲) Óleo de canola, (✶) Óleo de canola adulterado com óleo de soja e (■) Óleo de soja.

5.4.1 Resultados para os espectros de ^{1}H RMN

Os azeites apresentaram um agrupamento separado dos demais óleos por meio da PC1, embora possuam como origem diferentes países (Espanha, Portugal e Argentina). No gráfico dos escores construídos a partir de um modelo PCA constituídos das amostras de azeites (Anexo 3.1), foi possível observar que amostras argentinas de azeite não apresentaram separação pela PC1 ou PC2, mostrando que um conjunto maior de amostras seria necessário para capturar a variância necessária para separação de amostras de origens geográficas distintas.

O azeite contém uma alta quantidade de ácido oleico, um baixo nível de ácido linoleico e um nível muito baixo de ácido linolênico (TRADE STANDARD APPLYING TO OLIVE OILS AND OLIVE POMACE OILS, 2019). Os óleos de soja e canola apresentaram teores de ácido linoleico e linolênico consideravelmente altos, muito além dos valores normais encontrados no azeite. Assim, esses dois ácidos graxos, principalmente o ácido linolênico, poderiam ser utilizados como parâmetro para a detecção de fraudes na mistura desses óleos no (JAFARI; KADIVAR; KERAMAT, 2009; POPESCU *et al.*, 2015). Contudo, mesmo com a diferença de composição, as amostras de azeite e amostras de azeite adulterada com 5% de óleo se soja se agrupam.

Ambas as classes de amostras adulteradas em 5% de óleo de soja em volume apresentaram pouca distinção da respectiva classe sem adulteração. Isso indica que, para um conjunto de dados desse tamanho, a adulteração não poderia ser identificada pelo PCA para essa proporção de mistura (Figura 20). Foram testados modelos com três componentes principais, mas mesmo assim nenhuma melhora foi observada.

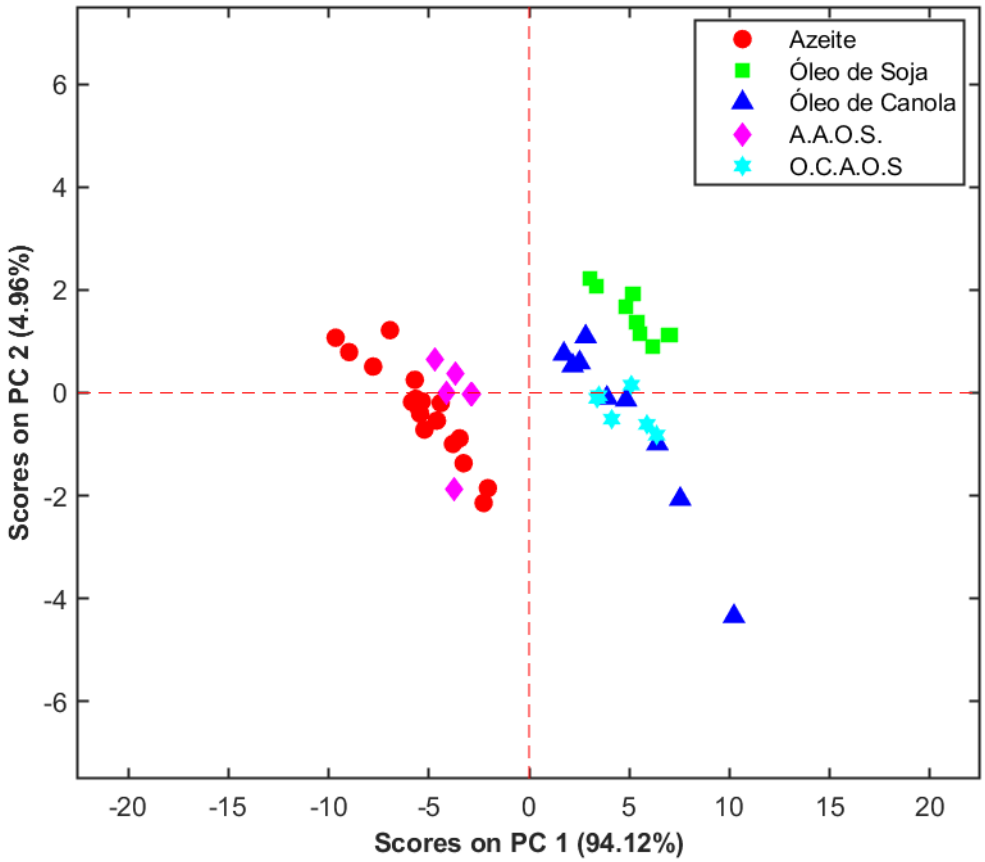


Figura 21. Gráficos bidimensionais do modelo PCA para espectros dos óleos vegetais de ^{1}H. (♦) A.A.O.S. Azeite adulterado com óleo de soja, (✶) O.C.A.O.S. Óleo de canola adulterado com óleo de soja

Através dos gráficos de pesos (*Loadings*) de cada componente principal, foi possível observar quais sinais espectrais são responsáveis pela distribuição das amostras nos gráficos dos escores e, dessa forma, associar a separação entre as classes pela composição das amostras.

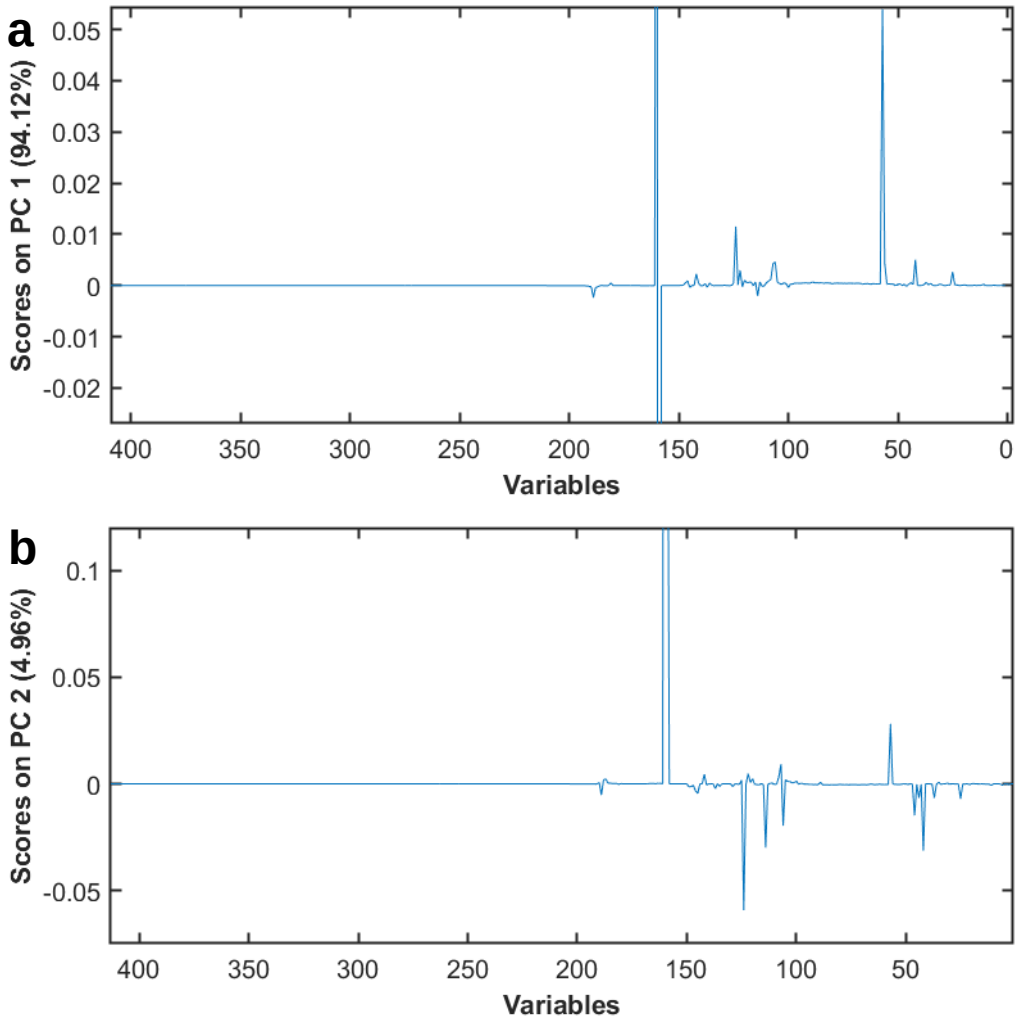


Figura 22. Gráfico de pesos dos espectros de RMN de ^{1}H com pré-processamento mean center para a) PC1 e b) PC2

O gráfico de pesos da PC1 (Figura 21a) revelou a presença de variáveis importantes para a separação das espécies na região de maior sobreposição de sinais, as quais estão em deslocamentos químicos semelhantes aos observados em PC2. Nas figuras 21a e 22b, as variáveis que mais contribuem no deslocamento dos grupos de amostras nos gráficos de escores estão presentes nos *bins* 160 e 136, região característica entre 2,10 – 1,90 ppm e 2,35 – 2,20 ppm, respectivamente. Na PC1, um *bin* com valor positivo responsável pela separação das classes presente em 136 é equivalente ao deslocamento químico em 2,80–2,70 ppm. Na PC2, o sinal do *bin* em 107 representa o deslocamento em 1,40 – 1,15 ppm, sendo essa região no espectro repleta de sinais sobrepostos para todas as amostras.

5.4.2 Resultados para os espectros de ^{13}C RMN

O gráfico de dispersão PC1 x PC2 para os espectros de RMN de ^{13}C mostrou a formação de clusters semelhantes aos observados para os espectros de RMN de ^{1}H para as amostras de azeite e óleo de Soja. As amostras de óleo de Canola apresentaram uma menor separação das amostras de Azeite pela PC1 quando comparado aos espectros de RMN de ^{1}H. Os componentes principais PC1 e PC2, mostrados no gráfico de dispersão da figura 22, explicaram 92,95% da variância total (PC1 81,64%, PC2 11,28%) enquanto PC3 (3,65%) e PC4 (2,09%) corresponderam apenas por aproximadamente 6%.

Os óleos de canola adulterados foram quase todos incluídos na faixa entre - 0,10 e 0,39 para a PC1, mesma faixa de distribuição para as amostras sem adulteração. Foi observada para esse conjunto de espectros que não foi possível a separação entre os óleos de referência e as adulterações preparadas com 5% de óleo de soja por meio da PC1 ou pela PC2. O mesmo foi observado para as amostras de azeite adulteras e sem adulteração.

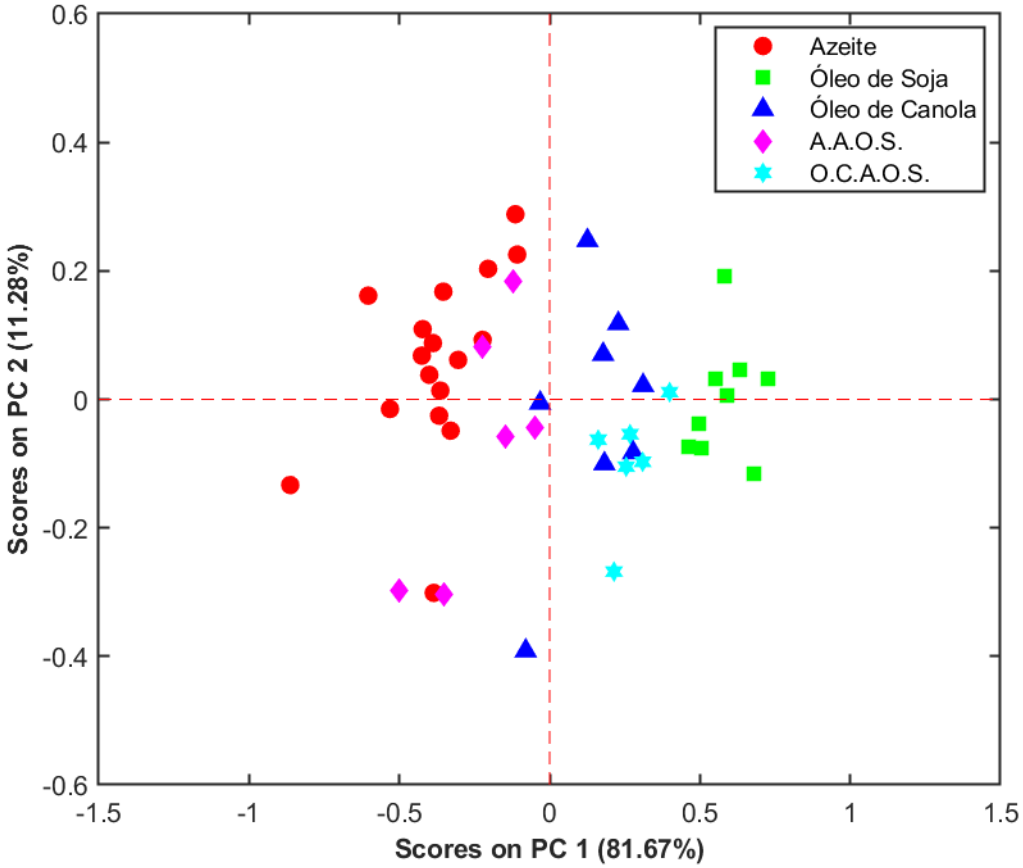


Figura 23. Gráficos bidimensionais do modelo PCA para espectros dos óleos vegetais de ^{13}C. (♦) A.A.O.S. Azeite adulterado com óleo de soja, (✶) O.C.A.O.S. Óleo de canola adulterado com óleo de soja

Dois grupos foram observados para as amostras de óleo de Canola e óleo de Soja ao longo do eixo PC1, o maior conjunto de amostras (14 amostras de óleo de Canola + adulterados), entre -0,10 e 0,39, enquanto o segundo grupo (9 amostras de óleo de Soja) caiu em uma faixa entre 0,45 e - 0,70. Houve uma clara separação entre esse conjunto de amostras.

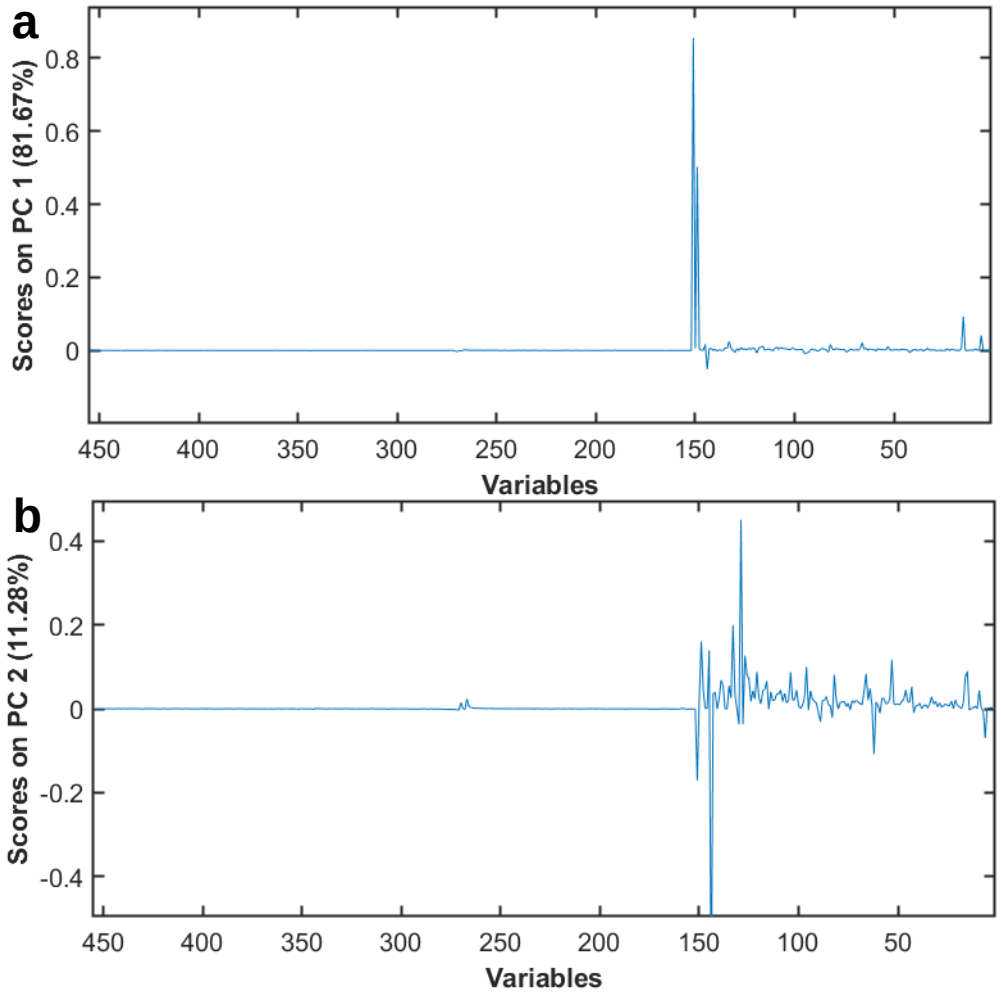


Figura 24. Gráfico de pesos dos espectros de RMN de ^{13}C com pré-processamento mean center para a) PC1 e b) PC2

A inspeção da PC1 e PC2 dos pesos na figura 23a e 23b revelou que os deslocamentos químicos observados nos *bins* em 149 e 151 (PC1) e 129, 144, 149 e 151 (PC1) parecem ter maior influência na separação observada na primeira e segunda componente principal nos gráficos de escores. Essas regiões delimitadas pelos bins são referentes aos deslocamentos químicos em: 25,56 – 25,72 (*bin* 129), 27,22 – 27,38 (*bin* 144), 27,22 – 27,38 (*bin* 149) e 29,42 (*bin* 151). Esses deslocamentos sugerem que o teor de ácido oleico e linoleico tem uma maior influência na variação observada ao longo dos dois primeiros componentes principais nos gráficos dos escores.

Por fim, como apresentado na figura 24, o modelo construído com base nos parâmetros selecionados pelo usuário – número de PCs, bin e *slackness*, número de amostras, variância explicada, etc – podem ser salvos e exportados usando o painel de exportação, o mesmo para as figuras salvas na visualização dos resultados, como exemplo o gráfico bidimensional do modelo PCA apresentado na Figura 22. Se necessário, o modelo com todas as informações pode ser carregado novamente na guia principal do PCA, usando o botão Carregar modelo (*Loading Model*).

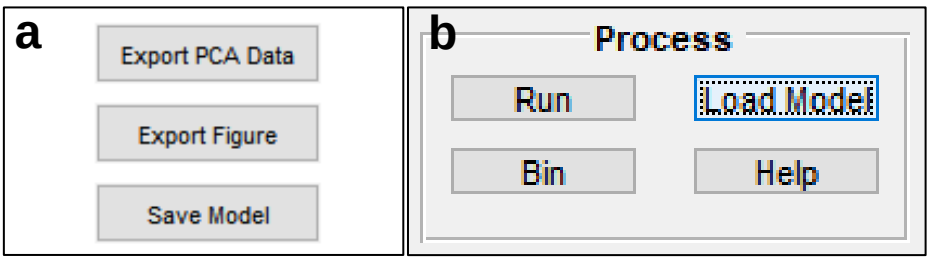


Figura 25. a) Painel para exportar as informações visuais na PCA GUI e o modelo PCA calculado e b) Painel inicial no GNAT para carregamento do modelo salvo

Como forma de validar os cálculos e resultados na interface gráfica desenvolvida, os dados pré-processados pelo GNAT foram salvos e analisados pelo Software PLS-Toolbox, os quais apresentaram resultados de escores e pesos equivalentes aos obtidos com o GNAT. A título de comparação, os gráficos de pesos e escores obtidos pelo PLS-Toolbox são apresentados no Anexo 3.2.

6 CONCLUSÃO

Nesse trabalho foram desenvolvidas funções quimiométricas dentro do ambiente do MATLAB ao pacote computacional do GNAT. As funções fundamentais para construção do modelo PCA e para a visualização dos resultados foram validadas comparando os resultados com o software pago PLS-Toolbox. O cálculo dos limites dos *bins* otimizado e do método convencional foi implementado como pré-processamento primário para construção desse modelo.

Por meio dos dados espectrais de RMN de 1H e ^{13}C de amostras de óleo de oliva, soja e canola obtidos no instituto de química da UNICAMP, São Paulo, foi possível demonstrar a implementação do PCA no GNAT, assim como avaliar as ferramentas do software desenvolvidas para interpretação desses dados. O gráfico de dispersão dos escores PC1 x PC2 para os espectros de RMN de ^{13}C e ^{1}H mostrou a formação de grupos separados para as amostras de azeite e óleo de soja e canola majoritariamente pela PC1. Esses deslocamentos sugerem que o teor de ácido linoleico e linolênico, têm uma maior influência na variação observada ao longo dos dois primeiros componentes principais nos gráficos dos escores.

Como perspectiva futura, sugere-se continuar dando o suporte à plataforma, realizando correções necessárias nas funções do programa de acordo com o retorno dos usuários por meio do e-mail mathias.nilsson@manchester.ac.uk. Assim como implementar outras ferramentas quimiométricas necessárias para complementar o número de funções na plataforma. Além disso, pretende-se transportar essas funções para outra plataforma do grupo de Manchester para lidar com dados de RMN em dimensões superiores, MAGNATE (Multidimensional Analysis for the GNAT Environment).

7 REFERÊNCIAS BIBLIOGRÁFICAS

ACCESS MATLAB ADD-ON TOOLBOXES - MATLAB & SIMULINK. [*S. l.*], [*s. d.*]. Disponível em: https://www.mathworks.com/help/thingspeak/matlab-toolbox-access.html. Acesso em: 20 abr. 2022.

ALONSO-SALCES, Rosa M.; HOLLAND, Margaret V.; GUILLOU, Claude. 1H-NMR fingerprinting to evaluate the stability of olive oil. **Food Control**, [*s. l.*], v. 22, n. 12, p. 2041–2046, 2011. Disponível em: http://dx.doi.org/10.1016/j.foodcont.2011.05.026.

ANDERSON, Paul E. *et al.* Dynamic adaptive binning: An improved quantification technique for NMR spectroscopic data. **Metabolomics**, [*s. l.*], v. 7, n. 2, p. 179–190, 2011.

ANDERSON, Paul E. *et al.* Gaussian binning: A new kernel-based method for processing NMR spectroscopic data for metabolomics. **Metabolomics**, [*s. l.*], v. 4, n. 3, p. 261–272, 2008.

BARISON, Andersson *et al.* A simplemethodology for the determination of fatty acid composition in edible oils through 1H NMR spectroscopy. **Magnetic Resonance in Chemistry**, [*s. l.*], v. 48, n. 8, p. 642–650, 2010.

BARTHOLOMEW, D. J. Principal components analysis. **International Encyclopedia of Education**, [*s. l.*], p. 374–377, 2010.

BRESCIA, Maria Antonietta *et al.* Chemometric Classification of Olive Cultivars Based on Compositional Data of Oils. **JAOCS, Journal of the American Oil Chemists' Society**, [*s. l.*], v. 80, n. 10, p. 945–950, 2003.

CASTAÑAR, Laura *et al.* The GNAT: A new tool for processing NMR data. **Magnetic Resonance in Chemistry**, [*s. l.*], v. 56, n. 6, p. 546–558, 2018.

CHEMOMETRICS - DATA ANALYSIS SOFTWARE - PLS_TOOLBOX - EIGENVECTOR. [*S. l.*], [*s. d.*]. Disponível em: https://eigenvector.com/software/pls-toolbox/. Acesso em: 20 abr. 2022.

CHRISTY, Alfred A. *et al.* The detection and quantification of adulteration in olive oil by near-infrared spectroscopy and chemometrics. **Analytical Sciences**, [*s. l.*], v. 20, n. 6, p. 935–940, 2004.

CICHELLI, Angelo; PERTESANA, Gian Pietro. High-performance liquid chromatographic analysis of chlorophylls, pheophytins and carotenoids in virgin olive oils: Chemometric approach to variety classification. **Journal of Chromatography A**, [*s. l.*], v. 1046, n. 1–2, p. 141–146, 2004.

COBAS, Juan Carlos; SARDINA, F. Javier. Nuclear magnetic resonance data processing. MestRe-C: A Software package for desktop computers. **Concepts in Magnetic Resonance Part A: Bridging Education and Research**, [*s. l.*], v. 19, n. 2, p. 80–96, 2003.

CRAIG, Edward C.; MARSHALL, Alan G. Automated phase correction of FT NMR spectra by means of phase measurement based on dispersion versus absorption relation (DISPA). **Journal of Magnetic Resonance (1969)**, [*s. l.*], v. 76, n. 3, p. 458–475, 1988.

DAOLIO, Cristina *et al.* Classification of commercial Catuaba samples by NMR, HPLC and chemometrics. **Phytochemical Analysis**, [*s. l.*], v. 19, n. 3, p. 218–228, 2008.

DAVIS, Richard A. *et al.* Adaptive binning: An improved binning method for metabolomics

data using the undecimated wavelet transform. **Chemometrics and Intelligent Laboratory Systems**, [*s. l.*], v. 85, n. 1, p. 144–154, 2007.

DI PIETRO, Maria Enrica; MANNU, Alberto; MELE, Andrea. NMR determination of free fatty acids in vegeTabela oils. **Processes**, [*s. l.*], v. 8, n. 4, 2020.

FAUHL, Carsten; RENIERO, Fabiano; GUILLOU, Claude. 1H NMR as a tool for the analysis of mixtures of virgin olive oil with oils of different botanical origin. **Magnetic Resonance in Chemistry**, [*s. l.*], v. 38, n. 6, p. 436–443, 2000.

FORSHED, Jenny *et al.* A comparison of methods for alignment of NMR peaks in the context of cluster analysis. **Journal of Pharmaceutical and Biomedical Analysis**, [*s. l.*], v. 38, n. 5 SPEC. ISS., p. 824–832, 2005.

GÜNTERT, Peter *et al.* Processing of multi-dimensional NMR data with the new software PROSA. **Journal of Biomolecular NMR**, [*s. l.*], v. 2, n. 6, p. 619–629, 1992.

HARROU, Fouzi *et al.* Improved principal component analysis for anomaly detection: Application to an emergency department. **Computers and Industrial Engineering**, [*s. l.*], v. 88, p. 63–77, 2015. Disponível em: http://dx.doi.org/10.1016/j.cie.2015.06.020.

HÉBERGER, Károly; RAJKÓ, Róbert. Generalization of pair correlation method (PCM) for non-parametric variable selection. **Journal of Chemometrics**, [*s. l.*], v. 16, n. 8–10, p. 436–443, 2002.

HOTELLING, Harold. Analysis of a complex of statistical variables into Principal Components. Jour. Educ. Psych., 24, 417-441, 498-520. **The Journal of Educational Psychology**, [*s. l.*], v. 24, p. 417–441, 1933.

JACKSON, J. Edward; MUDHOLKAR, Govind S. Control procedures for residuals associated with principal component analysis. **Technometrics**, [*s. l.*], v. 21, n. 3, p. 341–349, 1979.

JAFARI, Maryam; KADIVAR, Mahdi; KERAMAT, Javad. Detection of adulteration in Iranian olive oils using instrumental (GC, NMR, DSC) methods. **JAOCS, Journal of the American Oil Chemists' Society**, [*s. l.*], v. 86, n. 2, p. 103–110, 2009.

JAKAB, Annamaria; HÉBERGER, Károly; FORGÁCS, Esther. Comparative analysis of different plant oils by high-performance liquid chromatography-atmospheric pressure chemical ionization mass spectrometry. **Journal of Chromatography A**, [*s. l.*], v. 976, n. 1–2, p. 255–263, 2002.

KEIFER, Paul A. Flow NMR applications in combinatorial chemistry. **Current Opinion in Chemical Biology**, [*s. l.*], v. 7, n. 3, p. 388–394, 2003.

KOURTI, Theodora; MACGREGOR, John F. Process analysis, monitoring and diagnosis, using multivariate projection methods. **Chemometrics and Intelligent Laboratory Systems**, [*s. l.*], v. 28, n. 1, p. 3–21, 1995.

LARSEN, Flemming H.; VAN DEN BERG, Frans; ENGELSEN, Søren B. An exploratory chemometric study of 1H NMR spectra of Tabela wines. **Journal of Chemometrics**, [*s. l.*], v. 20, n. 5, p. 198–208, 2006.

LENZ, E. M. *et al.* A 1H NMR-based metabonomic study of urine and plasma samples obtained from healthy human subjects. **Journal of Pharmaceutical and Biomedical Analysis**, [*s. l.*], v. 33, n. 5, p. 1103–1115, 2003.

LIA, Frederick *et al.* Application of 1H and 13C NMR fingerprinting as a tool for the authentication of maltese extra virgin olive oil. **Foods**, [*s. l.*], v. 9, n. 6, p. 1–14, 2020.

LUCASIUS, C. B.; KATEMAN, G. Understanding and using genetic algorithms Part 1. Concepts, properties and context. **Chemometrics and Intelligent Laboratory Systems**, [*s. l.*], v. 19, n. 1, p. 1–33, 1993.

MACGREGOR, J. F.; KOURTI, T. Statistical process control of multivariate processes. **Control Engineering Practice**, [*s. l.*], v. 3, n. 3, p. 403–414, 1995.

MAGRITEK | BENCHTOP NMR PERFORMANCE AND QUALITY. [*S. l.*], [*s. d.*]. Disponível em: https://magritek.com/. Acesso em: 20 abr. 2022.

MANNINA, L; SOBOLEV, a P; A., Segre. Olive oil as seen by NMR and chemometrics. **Spectroscopy Europe**, [*s. l.*], v. 15, n. 3, p. 6–14, 2003.

MUJICA, L. E. *et al.* Q-statistic and t2-statistic pca-based measures for damage assessment in structures. **Structural Health Monitoring**, [*s. l.*], v. 10, n. 5, p. 539–553, 2011.

NIELSEN, Niels Peter Vest; CARSTENSEN, Jens Michael; SMEDSGAARD, Jørn. Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. **Journal of Chromatography A**, [*s. l.*], v. 805, n. 1–2, p. 17–35, 1998.

NMR SPECTROSCOPY SOFTWARE FROM ACD/LABS. [*S. l.*], [*s. d.*]. Disponível em: https://www.acdlabs.com/products/adh/nmr/. Acesso em: 20 abr. 2022.

OLIVERI, P. *et al.* Application of class-modelling techniques to near infrared data for food authentication purposes. **Food Chemistry**, [*s. l.*], v. 125, n. 4, p. 1450–1456, 2011. Disponível em: http://dx.doi.org/10.1016/j.foodchem.2010.10.047.

PEARSON, Gerald A. A general baseline-recognition and baseline-flattening algorithm. **Journal of Magnetic Resonance (1969)**, [*s. l.*], v. 27, n. 2, p. 265–272, 1977.

POPESCU, Raluca *et al.* Discrimination of vegeTabela oils using NMR spectroscopy and chemometrics. **Food Control**, [*s. l.*], v. 48, p. 84–90, 2015. Disponível em: http://dx.doi.org/10.1016/j.foodcont.2014.04.046.

PRAVDOVA, V.; WALCZAK, B.; MASSART, D. L. A comparison of two algorithms for warping of analytical signals. **Analytica Chimica Acta**, [*s. l.*], v. 456, n. 1, p. 77–92, 2002.

QIN, S. Joe. Statistical process monitoring: basics and beyond. **Journal of Chemometrics**, [*s. l.*], v. 17, n. 8–9, p. 480–502, 2003. Disponível em: https://onlinelibrary.wiley.com/doi/full/10.1002/cem.800. Acesso em: 20 abr. 2022.

REZZI, Serge *et al.* Classification of olive oils using high throughput flow 1H NMR fingerprinting with principal component analysis, linear discriminant analysis and probabilistic neural networks. **Analytica Chimica Acta**, [*s. l.*], v. 552, n. 1–2, p. 13–24, 2005.

SAVORANI, F.; TOMASI, G.; ENGELSEN, S. B. icoshift: A versatile tool for the rapid alignment of 1D NMR spectra. **Journal of Magnetic Resonance**, [*s. l.*], v. 202, n. 2, p. 190–202, 2010.

SHAW, Adrian D. *et al.* Discrimination of the variety and region of origin of extra virgin olive oil using 13C NMR and multivariate calibration with variable reduction. **Analytica Chimica Acta**, [*s. l.*], v. 348, n. 1–3, p. 357–374, 1997.

SKOV, Thomas *et al*. Automated alignment of chromatographic data. **Journal of Chemometrics**, [*s. l.*], v. 20, n. 11–12, p. 484–497, 2006.

SOUSA, S. A.A.; MAGALHÃES, Alviclér; FERREIRA, Márcia Miguel Castro. Optimized bucketing for NMR spectra: Three case studies. **Chemometrics and Intelligent Laboratory Systems**, [*s. l.*], v. 122, p. 93–102, 2013.

TOMASI, Giorgio; VAN DEN BERG, Frans; ANDERSSON, Claus. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. **Journal of Chemometrics**, [*s. l.*], v. 18, n. 5, p. 231–241, 2004.

TOPSPIN | NMR DATA ANALYSIS | BRUKER. [*S. l.*], [*s. d.*]. Disponível em: https://www.bruker.com/en/products-and-solutions/mr/nmr-software/topspin.html?gclid=CjwKCAjwrqqSBhBbEiwAlQeqGtiQsWT6rBw-7W1o7tS0ayfJyOL39ZEtK5y55XKFCQ7IbJ_SzeGCfBoChmYQAvD_BwE. Acesso em: 20 abr. 2022.

TRADE STANDARD APPLYING TO OLIVE OILS AND OLIVE POMACE OILS. [*s. l.*], 2019. Disponível em: http://www.internationaloliveoil.org/. Acesso em: 20 abr. 2022.

TZOUROS, N. E.; ARVANITOYANNIS, I. S. Agricultural Produces: Synopsis of Employed Quality Control Methods for the Authentication of Foods and Application of Chemometrics for the Classification of Foods According to Their Variety or Geographical Origin. **http://dx.doi.org/10.1080/20014091091823**, [*s. l.*], v. 41, n. 4, p. 287–319, 2010. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/20014091091823. Acesso em: 20 abr. 2022.

WINNING, Hanne. Quantitative multivariate NMR spectroscopy in Food Science and Nutrition. [*s. l.*], p. 176, 2009.

8 ANEXOS

Algoritmo das Funções utilizadas no **Modulo I** – Visualização dos Resultados

- Anexo 1.1 - *pca_svd*

```
function model = pca_svd(X,ncomp,climit,varargin)
[m,n] = size(X);
mX = mean(X);
mtzX = repmat(mX,size(X,1),1);
meanMtx = (X - mtzX);
meanMtx2 = X - mtzX;

[u,s,v]= svd(meanMtx, 'econ');
Eigenvalue = diag(s).^2/(m-1);
Var_Exp = 100*((Eigenvalue)/sum(Eigenvalue));            % Explained
variance
Cum_Var_Exp = cumsum(Var_Exp);                           % Explained
variance accumulated
T = u*s;                                                 % Scores (T)
P = v;                                                   % Loadings (P)

Eigenvaluencomp = Eigenvalue(1:ncomp);
Var_Exp = Var_Exp(1:ncomp);
Cum_Var_Exp = Cum_Var_Exp(1:ncomp);
T = T(:,1:ncomp);
P = P(:,1:ncomp);

AxesPlot = {};
for i=1:ncomp
    AxesPlot{i,1} = sprintf('Scores on PC %d
(%4g%%)',i,round((Var_Exp(i,1)),2));
    AxesPlot{i,2} = sprintf('Loading on PC %d
(%4g%%)',i,round((Var_Exp(i,1)),3));
end

E = meanMtx - T*P';                                       % Residuals
Qres = [];
for i=1:size(T,1)
    Qres(i) = E(i,:)*E(i,:)';
end


I = zeros(size(T,2),size(T,2));                          % Hotelling's T.^2
for i=1:size(T,2)
    I(i,i) = Eigenvaluencomp(i);
end

T2 = [];
for i=1:size(T,1)
    T2(i) = T(i,:)*((I)\T(i,:)');
end

label = {};
for k=1:m
    label{k} = num2str(k);
end
```

```
Qlim  = qlimit(Eigenvalue,ncomp,climit);
T2lim = t2limit(m,ncomp,climit);
```

- Anexo 1.2 - *T2limit*

```
function T2limit = t2limit(m,ncomp,climit)
if license('test','statistics_toolbox')
    F = finv(climit,ncomp,m-ncomp);
    T2limit = ncomp*(m - 1)/(m - ncomp)*F;
else
    T2limit = NaN;
end
```

- Anexo 1.3 - *qlimit*

```
function Qalpha = qlimit(eigenvalue,ncomp, climit)
theta1 = sum(eigenvalue(ncomp+1:end));
theta2 = sum(eigenvalue(ncomp+1:end).^2);
theta3 = sum(eigenvalue(ncomp+1:end).^3);

h0 = 1-(2*theta1*theta3)/(3*(theta2^2));
if h0<0.001
    h0 = 0.001;
end

ca = sqrt(2)*erfinv(2*(climit-1)); % Is this correct? Have to check
a = (ca*sqrt(2*theta2*(h0.^2)))/theta1;
b = theta2*h0*(1-h0)/(theta1.^2);
Qalpha = theta1*(a + 1 + b).^1/h0;
```

Algoritmo das Funções utilizadas no **Modulo II** – Visualização dos Resultados

- Anexo 2.1 - *ScoreCheck_Callback* (Somente para plot unico)

```
function ScoreCheck_Callback(source,eventdata)
        PLOTguiData = guidata(hPLOTfigure);
        labels = string(1:ChemoData.arraydim);
        labels(ChemoData.prune) = [];
        Opt=ChemoData.parameters.PlotOpt;
        X=sum(ChemoData.parameters.PlotOpt);
        if X==1
            axes(Axes);
            if Opt(1) == 1 % score plot
                if get(hCheckUse3D,'value')
                    set(Axes,'Visible','off');
                    PLOTguiData.Axes.h =
scatter3(ChemoData.parameters.Scores(:,(get(hpopupPCX,'value'))),...

ChemoData.parameters.Scores(:,(get(hpopupPCY,'value'))),...

ChemoData.parameters.Scores(:,(get(hpopupPCZ,'value'))),...
                    'MarkerEdgeColor','k','MarkerFaceColor',[0 .75 .75]);
                    axis('tight');
                    set(PLOTguiData.Axes.h,'LineWidth',1);
                    set(gca,'LineWidth',1);
                    set(gca,'Box','on');
```

```
                    set(Axes,'XLim',[-max(abs(get(gca,'XLim'))).*1.5
max(abs(get(gca,'XLim'))).*1.5]);
            set(Axes,'YLim',[-max(abs(get(gca,'YLim'))).*1.5
max(abs(get(gca,'YLim'))).*1.5]);
            set(Axes,'ZLim',[-max(abs(get(gca,'ZLim'))).*1.5
max(abs(get(gca,'ZLim'))).*1.5]);

xlabel(Axes,ChemoData.parameters.AxesPlot((get(hpopupPCX,'value')),1),'Font
Size',10, 'FontWeight','bold');

ylabel(Axes,ChemoData.parameters.AxesPlot((get(hpopupPCY,'value')),1),'Font
Size',10, 'FontWeight','bold');

zlabel(Axes,ChemoData.parameters.AxesPlot((get(hpopupPCZ,'value')),1),'Font
Size',10, 'FontWeight','bold');
                    line([min(xlim) max(xlim)],[0 0],'color',[1 0
0],'linestyle','--');
                    line([0 0],[min(ylim) max(ylim)],'color',[1 0
0],'linestyle','--');
                    line([0 0],[0 0],[min(zlim) max(zlim)],'color',[1 0
0],'linestyle','--');
%                     labels = string(1:pca.parameters.samples);
                    ChemoData.labelScore =
labelpoints(ChemoData.parameters.Scores(:,(get(hpopupPCX,'value'))),ChemoDa
ta.parameters.Scores(:,(get(hpopupPCY,'value'))), labels,'N',0.05,1,
'FontSize', 10);
%labels = string(1:pca.parameters.samples);
%pca.label = text(pca.parameters.Scores(:,(get(hpopupPCX,'value'))),...
%
pca.parameters.Scores(:,(get(hpopupPCY,'value'))),...
%pca.parameters.Scores(:,(get(hpopupPCZ,'value'))) +0.02,labels);
                else

PLOTguiData.Axes.h = …,
gscatter(ChemoData.parameters.Scores(:,(get(hpopupPCX,'value'))),ChemoData.
parameters.Scores(:,(get(hpopupPCY,'value'))),...
                        Classes,'rgbmcyb','os^dhv',6,'on');
                    colors = 'rgbmcyb';
                    specxlim1=xlim(Axes);
                    specylim1=ylim(Axes);
                    assignin('base','specxlim1',specxlim1);
                    assignin('base','specylim1',specylim1);
                    for n = 1:length(PLOTguiData.Axes.h)
              set(PLOTguiData.Axes.h(n,1), 'MarkerFaceColor', colors(n));
                    end

%                     set(Axes, 'YLimSpec', 'stretch');
%                     axis equal

                    set(Axes,'XLim',[-max(abs(get(gca,'XLim'))).*1.5
max(abs(get(gca,'XLim'))).*1.5]);
                    set(Axes,'YLim',[-max(abs(get(gca,'YLim'))).*1.5
max(abs(get(gca,'YLim'))).*1.5]);

                    set(PLOTguiData.Axes.h,'LineWidth',1);
                    set(Axes,'LineWidth',1);
                    set(Axes,'Box','on');

xlabel(Axes,ChemoData.parameters.AxesPlot((get(hpopupPCX,'value')),1),'Font
Size',10, 'FontWeight','bold');
```

```
ylabel(Axes,ChemoData.parameters.AxesPlot((get(hpopupPCY,'value')),1),'Font
Size',10, 'FontWeight','bold');
                    line([min(xlim) max(xlim)],[0 0],'color',[1 0
0],'linestyle','--');
                    line([0 0],[min(ylim) max(ylim)],'color',[1 0
0],'linestyle','--');
%                     labels = string(1:pca.parameters.samples);
%                     ChemoData.labelScore =
labelpoints(ChemoData.parameters.Scores(:,(get(hpopupPCX,'value'))),ChemoDa
ta.parameters.Scores(:,(get(hpopupPCY,'value'))), labels,'N',0.05,1,
'FontSize', 10);
%                     labels = string(1:pca.parameters.samples);
%                     pca.label =
text(pca.parameters.Scores(:,(get(hpopupPCX,'value'))),...
%                         pca.parameters.Scores(:,(get(hpopupPCY,'value')))
+
2.*(abs(mean((pca.parameters.Scores(:,(get(hpopupPCY,'value')))))))),labels)
;
                end
            elseif Opt(2) == 1 % loading plot
                if isfield(ChemoData,'BinPCA')

PLOTguiData.Axes.h=scatter(ChemoData.parameters.Loadings(:,(get(hpopupPCXLo
ading,'value'))),ChemoData.parameters.Loadings(:,(get(hpopupPCYLoading,'val
ue'))));
                    axis('tight');
                    set(PLOTguiData.Axes.h,'LineWidth',1);
                    set(gca,'LineWidth',1);
                    set(gca,'xdir','reverse');
                    set(Axes,'YLim',[-max(abs(get(gca,'YLim'))).*1.1
max(abs(get(gca,'YLim'))).*1.1]);
                    xlabel('Variables','FontSize',10, 'FontWeight','bold');

ylabel(ChemoData.parameters.AxesPlot((get(hpopupPCY,'value')),1),'FontSize'
,10, 'FontWeight','bold');
                    guidata(hPLOTfigure,PLOTguiData);
                    dcm_obj = datacursormode(hPLOTfigure);
                    assignin('base','dcm_obj',dcm_obj);
                    datacursormode on
%                     set(dcm_obj,'UpdateFcn',@myupdatefcn)

                    c_info = getCursorInfo(dcm_obj);
                    assignin('base','c_info',c_info);
                    % Make selected line wider
%                     set(c_info.Target,'LineWidth',5)
                else

PLOTguiData.Axes.h=plot(ChemoData.Ppmscale(1,:),ChemoData.parameters.Loadin
gs(:,(get(hpopupPCYLoading,'value'))));
                    axis('tight');
                    set(PLOTguiData.Axes.h,'LineWidth',1);
                    set(gca,'LineWidth',1);
                    set(gca,'xdir','reverse');
                    set(Axes,'YLim',[-max(abs(get(gca,'YLim'))).*1.1
max(abs(get(gca,'YLim'))).*1.1]);
                    xlabel('Chemical shift (ppm)','FontSize',10,
'FontWeight','bold');
```

```
ylabel(ChemoData.parameters.AxesPlot((get(hpopupPCYLoading,'value')),1),'Fo
ntSize',10, 'FontWeight','bold');
                end
            elseif Opt(3) == 1 % residuals plot
                    specxlim1=xlim(Axes);
                    specylim1=ylim(Axes);
                    assignin('base','specxlim1',specxlim1);
                    assignin('base','specylim1',specylim1);
                PLOTguiData.Axes.h =
gscatter(ChemoData.parameters.T2(1,:),ChemoData.parameters.Qres(1,:),Classe
s,'rgbmcyb','os^dhv',6,'off');
                colors = 'rgbmcyb';
                for n = 1:length(PLOTguiData.Axes.h)
                  set(PLOTguiData.Axes.h(n,1), 'MarkerFaceColor',
colors(n));
                end
                axis('tight');
                set(gca,'Box','on');
                set(PLOTguiData.Axes.h,'LineWidth',1);
                set(gca,'LineWidth',1);
                xlabel('Hotteling T²','FontSize',10, 'FontWeight','bold');
                ylabel('Q Residuals','FontSize',10, 'FontWeight','bold');
                if max(ChemoData.parameters.Qres) >
ChemoData.parameters.Qlim
                    line([ChemoData.parameters.T2lim
ChemoData.parameters.T2lim],[0
(max(ChemoData.parameters.Qres)).*1.25],'color',[1 0 0],'linestyle','--');
% T2 Limit
                else
                    line([ChemoData.parameters.T2lim
ChemoData.parameters.T2lim],[0
(ChemoData.parameters.Qlim).*1.25],'color',[1 0 0],'linestyle','--'); % T2
Limit
                end
                if max(ChemoData.parameters.T2) >
ChemoData.parameters.T2lim
                    line([0
(max(ChemoData.parameters.T2)).*1.25],[ChemoData.parameters.Qlim
ChemoData.parameters.Qlim],'color',[1 0 0],'linestyle','--'); % Q limit
                else
                    line([0
(ChemoData.parameters.T2lim).*1.25],[ChemoData.parameters.Qlim
ChemoData.parameters.Qlim],'color',[1 0 0],'linestyle','--'); % Q limit
                end
%                 labels = string(1:pca.parameters.samples);
                ChemoData.labelResidual =
labelpoints(ChemoData.parameters.T2(1,:),ChemoData.parameters.Qres(1,:),
labels,'N',0.05,1, 'FontSize', 10);
            end
        guidata(hPLOTfigure,PLOTguiData);
end
```

- Anexo 2.2 - *EnableZaxis_Callback*

```
function EnableZaxis_Callback(source,eventdata)
        if get(hCheckUse3D,'value')
            set(hpopupPCZ,'enable','on')
            ScoreCheck_Callback()
        else
```

```
            set(hpopupPCZ,'enable','off')
            ScoreCheck_Callback()
        end
end
```

- Anexo 2.3 - *ChangingScore_Callback*

```
function ChangingScore_Callback(source,eventdata)
     PLOTguiData = guidata(hPLOTfigure);
     ScoreCheck_Callback()
     guidata(hPLOTfigure,PLOTguiData);
end
```

- Anexo 2.4 - *EnableLabel_Callback*

```
function EnableLabel_Callback(source,eventdata)
        PLOTguiData = guidata(hPLOTfigure);
        if get(hCheckLabels,'value')
            if isfield(ChemoData,'labelScore')
                set(ChemoData.labelScore,'Visible','on')
            else
            end
            if isfield(ChemoData,'labelResidual')
                set(ChemoData.labelResidual,'Visible','on')
            else
            end
        else
            if isfield(ChemoData,'labelScore')
                set(ChemoData.labelScore,'Visible','off')
            else
            end
            if isfield(ChemoData,'labelResidual')
                set(ChemoData.labelResidual,'Visible','off')
            else
            end
        end
        guidata(hPLOTfigure,PLOTguiData);
end
```

Anexo 3.1 – Plot dos escores da PC1xPC2 para a classe de azeites no modelo PCA para um conjunto de dados com 18 amostras.

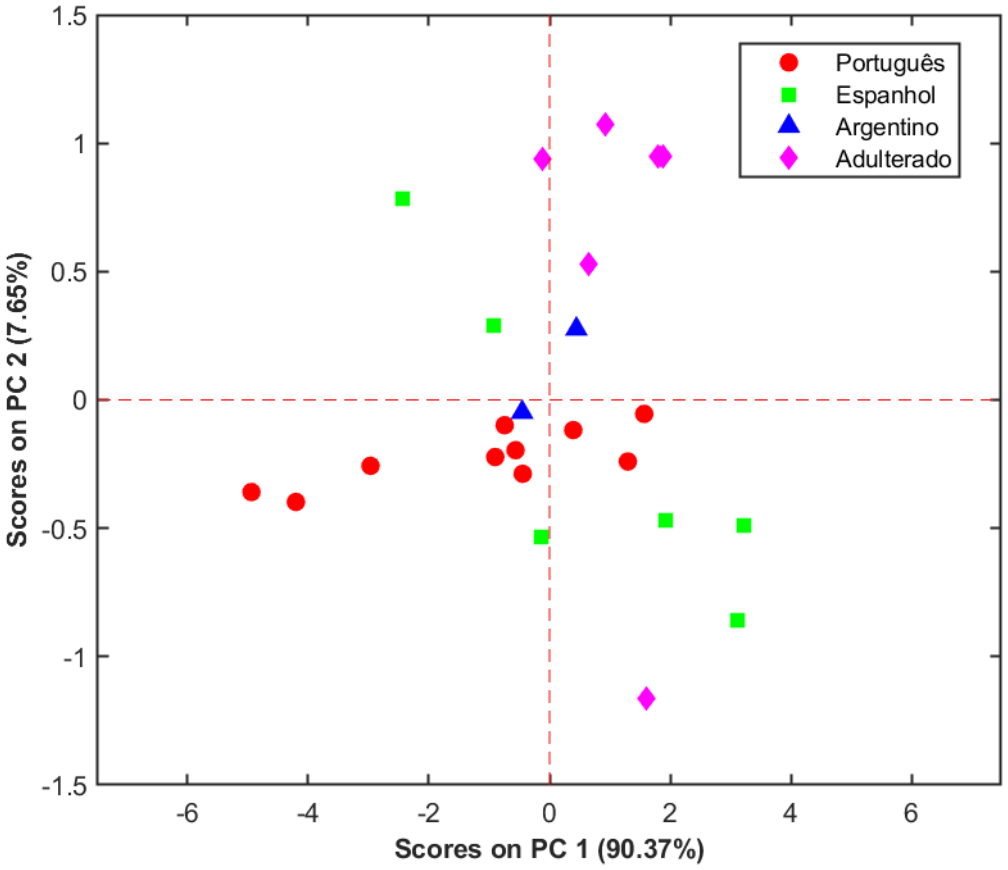


Figura 26. Gráfico dos escores construídos a partir de um modelo PCA constituídos utilizando somente as amostras de azeites

Anexo 3.2 – Resultado dos modelos PCA em diferentes plataformas.

a

Variance Captured by PCA | Cross-Validation | Class (Optional)

	Eigenvalue	Explained Variance (%)	Cumulative Variance (%)	RMSECV
1	28.4262	94.1246	94.1246	4.2402
2	1.4987	4.9626	99.0872	6.8074
3	0.1819	0.6022	99.6894	5.5837
4	0.0780	0.2582	99.9476	0

b

	Eigenvalue of Cov(X)	% Variance This PC	% Variance Cumulative	RMSEC	RMSECV	
1	2.84e+01	94.12	94.12	0.06486	0.3554	
2	1.50e+00	4.96	99.09	0.02557	0.3641	
3	1.82e-01	0.60	99.69	0.01491	0.5132	
4	7.80e-02	0.26	99.95	0.006126	0.1997	
5	1.32e-02	0.04	99.99	0.002491	0.09749	current*
6	1.43e-03	0.00	100.00	0.001676	0.08112	
7	4.27e-04	0.00	100.00	0.00134	0.08864	

Figura 27. Comparação dos resultados da variância explicada no modelo PCA construído no a) GNAT e no b) PLS_toolbox.