AnalyticalChemistryLake - Método para acessar, estruturar e armazenar dados analíticos a partir de técnicas de cromatografia

Luiz Eduardo D. Paiva[1], Giseli Rabello Lopes[1], Sergio Manuel Serra da Cruz[1]

[1]Programa de Pós-Graduação em Informática – Universidade Federal do Rio de Janeiro
Rio de Janeiro - RJ – Brasil

ledp@iq.ufrj.br, giseli@ic.ufrj.br, serra@ppgi.ufrj.br

Resumo. *Este trabalho apresenta um método computacional para acesso, estruturação e armazenamento de grandes volumes de dados analíticos gerados por técnicas cromatográfica, com foco em engenharia de dados e processamento analítico (OLAP). A proposta, denominada AnalyticalChemistryLake, visa superar limitações recorrentes em ambientes laboratoriais, tais como dependência de softwares proprietários, ausência de acesso programático aos dados, necessidade de exportações manuais e dificuldade de reprodutibilidade e automação das análises. O método baseia-se na conversão de arquivos brutos proprietários para o formato aberto mzML, seguida pela extração, modelagem e persistência dos dados em um repositório analítico estruturado.*

1. Introdução

O avanço das técnicas instrumentais em química analítica tem resultado na geração de grandes volumes de dados experimentais, caracterizando um cenário de *big data* científico em ambientes laboratoriais [Gleaves et al. 2021]. Esses dados são, em geral, armazenados em formatos proprietários e fortemente acoplados aos *softwares* fornecidos pelos fabricantes dos equipamentos, o que restringe o acesso programático, a integração entre fontes e a exploração sistemática das informações [Mogollon et al. 2014]. Embora esses sistemas sejam adequados para a operação rotineira e para a visualização básica dos resultados, eles impõem limitações quando se busca automação de análises, reprodutibilidade computacional e reutilização dos dados em fluxos de trabalho científicos [Mattoso et al. 2010].

Na prática, o acesso aos dados contidos nos arquivos brutos costuma restringir-se à visualização por meio de cromatogramas e espectros nas interfaces gráficas dos softwares proprietários, sem a possibilidade de extração direta e estruturada das informações subjacentes [François 2019]. Quando algum tipo de exportação é viável, ela depende de procedimentos manuais e específicos para cada fabricante, tornando o processo pouco escalável, suscetível a erros humanos e incompatível com pipelines automatizados de análise de dados [Gleaves et al. 2021].

Além disso, a dependência de ferramentas proprietárias compromete a transparência e a reprodutibilidade dos resultados, uma vez que o acesso às informações pode variar conforme versões de *software*, sistemas operacionais e disponibilidade de licenças [Mattoso et al. 2010]. Essa realidade é ilustrada na Figura 1, que representa o fluxo típico dos dados em ambientes laboratoriais, nos quais os arquivos gerados pelos

equipamentos permanecem armazenados em mídias físicas e diretórios locais, sem integração, estruturação analítica ou acesso programático sistemático. No contexto da ciência de dados, abordagens baseadas em *workflows* científicos, pipelines automatizados e armazenamento analítico (OLAP) têm sido amplamente empregadas para lidar com grandes volumes de dados, promovendo escalabilidade, rastreabilidade e exploração multidimensional das informações [Mattoso et al. 2010; Kimball e Ross 2013; Inmon 2016]. Sistemas OLAP, em particular, são projetados para consultas analíticas complexas, agregações e análise orientada à leitura intensiva, sendo amplamente utilizados em cenários de big data e apoio à tomada de decisão [Kimball e Ross 2013; Inmon 2016]. Tais fundamentos oferecem uma base conceitual adequada para o tratamento de dados científicos que demandam organização estruturada e análise eficiente.

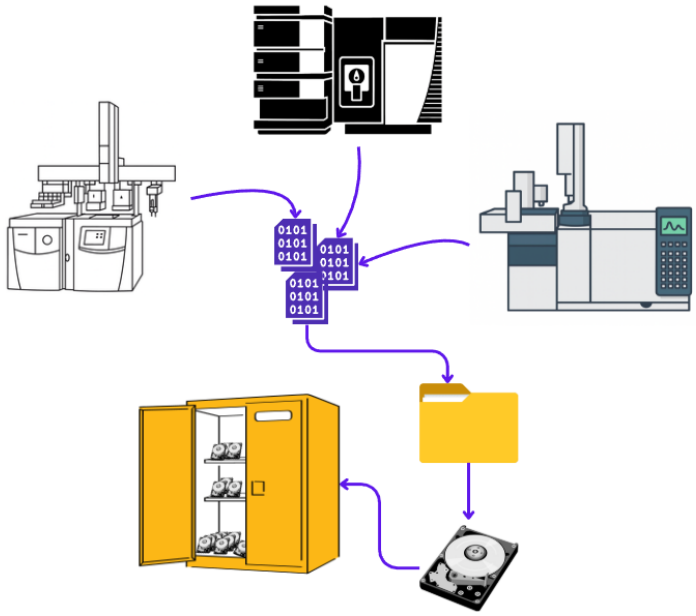

Figura 1. Fluxo típico de dados em ambientes laboratoriais analíticos.

Entretanto, apesar do potencial dessas abordagens, sua aplicação ainda é limitada em ambientes de química analítica, nos quais os dados permanecem fortemente dependentes de *softwares* proprietários e de fluxos de trabalho manuais [Mogollon et al. 2014]. Trabalhos recentes demonstram o valor de métodos computacionais no apoio à interpretação de dados cromatográficos e espectrométricos, seja por meio de pipelines de processamento ou de técnicas de ciência de dados e inteligência computacional [Gowd et al. 2018; Ryoo et al. 2024]. Contudo, tais estudos concentram-se majoritariamente na análise dos resultados, não abordando de forma sistemática a estruturação, persistência e reutilização dos dados experimentais em um repositório analítico unificado.

Diante desse cenário, este trabalho propõe o AnalyticalChemistryLake, um método computacional baseado em princípios de engenharia de dados e armazenamento analítico (OLAP) para o acesso, estruturação e armazenamento de grandes volumes de dados analíticos. A proposta fundamenta-se na conversão de arquivos brutos proprietários para formatos abertos, seguida pela modelagem e persistência dos dados em um

repositório analítico estruturado, com o objetivo de superar limitações relacionadas à dependência de *software* proprietário, à exportação manual e à baixa reprodutibilidade [Mattoso et al. 2010; Gleaves et al. 2021]. O método prioriza padronização, escalabilidade e reaproveitamento de estruturas de dados, características centrais de soluções modernas de engenharia de dados [Kimball e Ross 2013].

Ele foi desenvolvido e validado a partir de dois cenários experimentais: dados provenientes de LCHR-MS e de GC-MS em modo SIM, obtidos a partir de equipamentos de diferentes fabricantes. Essas técnicas são utilizadas neste trabalho apenas como casos de uso, com o objetivo de demonstrar que a abordagem não está acoplada a um equipamento, fabricante ou técnica específica, desde que os dados possam ser convertidos para formatos abertos suportados [François 2019]. Adicionalmente, este estudo avalia três estratégias de armazenamento analítico — DuckDB, PostgreSQL com TimescaleDB e ClickHouse — representativas de diferentes paradigmas de persistência de dados, a fim de identificar a alternativa mais adequada em termos de desempenho, escalabilidade e suporte a consultas analíticas [Kimball e Ross 2013; Inmon 2016]. Dessa forma, a principal contribuição deste trabalho consiste na proposição e avaliação de um método OLAP para a gestão estruturada de dados analíticos, estabelecendo uma base computacional independente de fabricante para a integração, automação e exploração de grandes volumes de dados laboratoriais.

2. Trabalhos Relacionados

Esta seção aborda trabalhos relacionados ao método proposto neste estudo, com ênfase em abordagens computacionais para acesso, estruturação, armazenamento e análise de grandes volumes de dados provenientes de espectrometria de massas. Em particular, são considerados trabalhos que empregam bancos de dados, formatos abertos e arquiteturas analíticas para viabilizar a organização e a reutilização de dados experimentais. A revisão prioriza estudos que exploram a integração de dados analíticos em repositórios estruturados, o uso de linguagens de consulta e a aplicação de princípios de data *warehouse*, *data lake* e processamento analítico (OLAP), estabelecendo o contexto técnico no qual o método proposto se insere.

Avanços mais recentes exploram o uso de bancos de dados relacionais e linguagens de consulta para viabilizar a análise retrospectiva e a reutilização de grandes volumes de dados analíticos. Mardal et al. [2023] propõem o ScreenDB, uma base de dados SQL para arquivamento de dados de LC-HRMS não direcionados, permitindo consultas rápidas sobre dezenas de milhares de arquivos ao longo de vários anos. O sistema armazena sinais iônicos deconvoluídos em estruturas tabulares, possibilitando monitoramento de desempenho instrumental, análise retrospectiva e identificação de novos alvos analíticos. Embora demonstre de forma clara o potencial do uso de SQL para análise escalável de dados analíticos, a abordagem é fortemente acoplada a um método instrumental específico e a um fluxo de processamento particular, limitando sua generalização para múltiplas técnicas, fabricantes ou esquemas de modelagem reutilizáveis.

Outras abordagens concentram-se na infraestrutura computacional para acesso eficiente a grandes volumes de dados brutos. Yao et al. [2015] apresentam o Metabolite Atlas, uma arquitetura que integra dados de LC/MS convertidos para o formato aberto

mzML com bancos de dados de alto desempenho, como o SciDB, permitindo consultas por *m/z* e tempo de retenção diretamente sobre os dados brutos. A proposta destaca a importância da conversão para formatos abertos, do acesso programático aos dados e da integração entre dados experimentais e metadados. Entretanto, o foco principal está na exploração interativa e na anotação de metabólitos, e não na definição de um método genérico de engenharia de dados orientado à persistência analítica, reaproveitamento estrutural e independência de técnica e fabricante.

Do ponto de vista conceitual, trabalhos sobre *Big Data* e *Data Lakes* discutem a necessidade de repositórios capazes de armazenar grandes volumes de dados heterogêneos em formato bruto, com posterior estruturação orientada à análise (*schema-on-read*). Miloslavskaya e Tolstoy [2016] caracterizam *data lakes* como repositórios escaláveis que mantêm dados em seu formato nativo, permitindo posterior integração com sistemas analíticos, incluindo abordagens OLAP. Esses conceitos são amplamente utilizados em contextos corporativos e científicos, mas sua aplicação sistemática ao domínio de dados analíticos laboratoriais ainda é pouco explorada, especialmente no que se refere à transformação de arquivos proprietários em formatos abertos, modelagem analítica e avaliação comparativa de tecnologias de armazenamento.

Em síntese, os trabalhos existentes demonstram avanços relevantes no uso de bancos de dados, formatos abertos e infraestrutura de alto desempenho para análise de dados de espectrometria de massas [Gaida e Neumann 2007; Yao et al. 2015; Mardal et al. 2023], bem como fundamentos conceituais para arquiteturas orientadas a grandes volumes de dados [Miloslavskaya e Tolstoy 2016]. Contudo, essas abordagens tendem a ser direcionadas a contextos específicos — seja à identificação de metabólitos, à exploração interativa ou à aplicação a um método instrumental particular — sem propor um método unificado de engenharia de dados, orientado à conversão para formatos abertos, modelagem reutilizável e armazenamento analítico (OLAP) independente de técnica, equipamento ou fabricante. Nesse contexto, o trabalho diferencia-se ao propor e avaliar um método estruturado para acesso, organização e persistência de dados analíticos em um repositório analítico, priorizando escalabilidade, reprodutibilidade e reaproveitamento de estruturas de dados.

3. Materiais e métodos

Este trabalho adota uma abordagem de engenharia de dados para acesso, estruturação e armazenamento analítico de dados cromatográficos, denominada AnalyticalChemistryLake. O método tem como objetivo eliminar a dependência de *softwares* proprietários, viabilizar o acesso programático aos dados e permitir sua organização em um repositório analítico, priorizando padronização, reprodutibilidade e escalabilidade. As técnicas instrumentais utilizadas (LCHR-MS e GC-MS em modo SIM) são empregadas exclusivamente como cenários experimentais para validação, não constituindo o foco metodológico do estudo.

Todo o fluxo de processamento é executado de forma automatizada por meio de um *script* em Python, responsável por integrar e orquestrar as etapas do método em um único *workflow*. Esse *script* realiza desde a identificação dos arquivos brutos no ambiente de aquisição até a extração, estruturação e persistência dos dados em um repositório

analítico, permitindo reexecução controlada do pipeline e reduzindo a necessidade de intervenção manual ao longo do processo.

Inicialmente, os dados são acessados diretamente a partir do ambiente de aquisição do equipamento, por meio do sistema de arquivos do *software* proprietário (por exemplo, ChemStation). Em vez de utilizar mecanismos de exportação manual, o método opera sobre os diretórios onde os arquivos brutos são armazenados, permitindo a automação da localização e do processamento de conjuntos de dados. Essa etapa garante que o fluxo de dados seja independente de interfaces gráficas e reduz a intervenção humana, favorecendo rastreabilidade e repetibilidade operacional.

Após o acesso aos arquivos brutos, o *script* aciona a ferramenta Msconvert, pertencente ao pacote ProteoWizard, para converter os dados para o formato aberto mzML. A conversão para um padrão aberto é um passo central do método, pois elimina o acoplamento a formatos proprietários e preserva tanto os dados experimentais quanto os metadados instrumentais. A Figura 2 apresenta o pipeline completo e evidencia a transição entre o ambiente proprietário e o fluxo de processamento independente de fabricante.

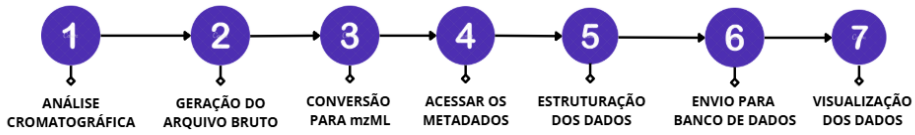

Figura 2. Pipeline do método AnalyticalChemistryLake.

Com o arquivo mzML gerado, o *script* utiliza a biblioteca Pyteomics para acessar programaticamente os dados e metadados. Nessa etapa, são extraídas informações necessárias para a reconstrução computacional do conteúdo original do arquivo bruto, incluindo parâmetros instrumentais, filtros de aquisição (*scan filters*), tempos de retenção, valores de *m/z*, intensidades, varreduras (*scans*) e estruturas espectrais. Esse processo assegura que a representação estruturada dos dados mantenha fidelidade ao conteúdo do experimento.

Em seguida, os dados extraídos são organizados em estruturas tabulares intermediárias, nas quais são definidos os elementos necessários para representar cromatogramas de massas específicas, picos, varreduras e espectros. O esquema de dados foi projetado para suportar dois cenários: (i) amostras que possuem informações espectrais associadas e (ii) amostras que contêm apenas dados cromatográficos. Para isso, os *scan filters* de cada amostra são identificados e armazenados de forma relacional, permitindo que registros com espectro sejam persistidos de maneira distinta daqueles que possuem somente cromatograma. Essa modelagem segue boas práticas de banco de dados, reduz redundâncias, mantém um número reduzido de tabelas e é definida uma única vez, podendo ser reutilizada para diferentes conjuntos de dados.

Após a definição do esquema lógico, as tabelas são criadas no banco de dados e o script realiza a carga dos dados estruturados no repositório analítico. O sistema de armazenamento adotado é o ClickHouse, um banco de dados orientado a colunas e projetado para cargas analíticas. A escolha do ClickHouse deve-se ao seu desempenho em consultas, à compressão eficiente dos dados e à adequação a cenários de leitura intensiva. Além disso, o uso do paradigma OLAP contribui para a preservação da

proveniência e integridade, uma vez que os dados, após inseridos no repositório, são utilizados prioritariamente para consulta e análise.

Por fim, os dados persistidos podem ser acessados por meio de consultas SQL, permitindo a recuperação de cromatogramas e espectros a partir de filtros por amostra, *scan filters*, intervalos de massa e tempo de retenção. A partir dessas consultas, são construídas visualizações que reproduzem computacionalmente os resultados contidos nos arquivos brutos, demonstrando que o método não apenas armazena os dados, mas também suporta sua exploração analítica e visualização sem dependência de softwares proprietários. A Figura 2 sintetiza o fluxo completo, desde a geração dos dados até sua análise no repositório.

4. Desenvolvimento

A implementação do método AnalyticalChemistryLake foi realizada por meio de um *script* único em Python, responsável por integrar e orquestrar todas as etapas do *workflow*: identificação dos arquivos brutos, conversão para formato aberto, extração e estruturação dos dados e persistência em um repositório analítico. Essa abordagem possibilita a execução automatizada e reprodutível do pipeline, reduzindo a intervenção manual e assegurando consistência no processamento de múltiplos conjuntos de dados. O *script* atua como o elemento central de controle do método, consolidando as operações de ingestão e análise em um fluxo contínuo.

Tabela 1. Funções e parâmetros utilizados no AnalyticalChemistryLak

Função / Modulo	Parâmetros principais	Descrição
find_inputs(base_dir)	base_dir	Varre recursivamente diretórios e identifica arquivos .raw e .D.
technique_from_input(p)	p	Determina a técnica (GC-MS ou LCHRMS) a partir da extensão do arquivo.
run_msconvert(input_path)	sample input_path, MSCONVERT, OVERWRITE_MZML	Executa o msconvert para gerar arquivos mzML a partir dos arquivos brutos.
insert_sample(sample_name)	sample_name	Cria registro da amostra no banco de dados.
insert_channel(sample_id, technique, scan_filter, sim_ion_name)	sample_id, technique, scan_filter, sim_ion_name	Cria canais associados à amostra, representando filtros de aquisição, íons SIM ou cromatogramas.

ingest_gcms_sim(mzml_path, sample_id)	mzml_path, sample_id	Processa cromatogramas GC-MS em modo SIM, extraindo tempo e intensidade por íon monitorado.
ingest_lcms_chromatograms(mzml_path, sample_id)	mzml_path, sample_id	Lê cromatogramas LC-MS do mzML e persiste pontos de tempo e intensidade.
ingest_lcms_fullscan_ms2(mzml_path, sample_id)	mzml_path, sample_id	Processa varreduras MS1 e MS2, armazenando scans e pontos espectrais (m/z, intensidade).
ch_insert_retry(table, rows, column_names)	table, rows, column_names, RETRY_DELAY	Insere dados em lotes no ClickHouse com mecanismo de repetição em caso de falha.
process_all(base_dir)	base_dir, DELETE_MZML_AFTER_INGEST	Orquestra o workflow completo: descoberta → conversão → ingestão → limpeza.

A etapa inicial do desenvolvimento concentra-se na descoberta automática dos arquivos de entrada no sistema de arquivos do ambiente de aquisição. Por meio de varredura recursiva de diretórios, o método identifica arquivos nos formatos proprietários suportados (.raw e .D) e infere a técnica associada com base na extensão do arquivo. Em seguida, a função de conversão aciona a ferramenta Msconvert (ProteoWizard) para transformar cada arquivo bruto em um arquivo mzML, estabelecendo a transição do ambiente proprietário para um formato aberto. Esse passo garante a independência de fabricante e preserva tanto os dados experimentais quanto os metadados necessários para as etapas subsequentes.

Após a conversão, o *script* utiliza a biblioteca Pyteomics para acessar programaticamente o conteúdo do mzML. Nesse ponto, são extraídos os metadados instrumentais, os filtros de aquisição (s*can filters*), os tempos de retenção, bem como os vetores de *m/z* e intensidade. A implementação diferencia os fluxos de ingestão conforme a técnica identificada: para GC-MS em modo SIM, são processados cromatogramas específicos por íon; para LCHR-MS, são ingeridos tanto cromatogramas quanto varreduras espectrais (MS e MS2), com tratamento distinto para registros que contêm espectros e para aqueles que apresentam apenas dados cromatográficos. Essa separação operacional permite acomodar diferentes estruturas de dados mantendo um esquema unificado de persistência.

A modelagem do banco de dados foi projetada para refletir a organização lógica dos dados analíticos com um número reduzido de tabelas e relacionamentos explícitos. Cada amostra é registrada uma única vez e associada a múltiplos canais, que representam filtros de aquisição, íons monitorados ou identificadores de cromatogramas. Para LCHR-MS, os canais organizam sequências de *scans* indexados, enquanto os pontos espectrais (*m/z*, intensidade) são vinculados a cada varredura; para GC-MS em SIM, os pontos cromatográficos (tempo, intensidade) são associados diretamente ao canal

correspondente ao íon monitorado. Essa estrutura evita redundâncias, facilita consultas analíticas e preserva a semântica necessária para reconstrução de cromatogramas e espectros.

A carga dos dados no repositório foi implementada por meio de inserções em lotes (*batching*), estratégia adotada para lidar com o elevado volume de pontos espectrais e, principalmente, para contornar limitações de memória RAM no ambiente de hospedagem do ClickHouse. Durante a ingestão, o envio de grandes volumes de dados em uma única operação pode ocasionar indisponibilidade temporária de memória; para evitar falhas e garantir a integridade do processo, o *script* fragmenta os dados em blocos controlados, enviando-os sequencialmente ao banco. Caso ocorram erros de inserção, um mecanismo de repetição é acionado, aguardando a liberação de recursos antes de prosseguir. Essa lógica assegura robustez operacional, evita perda de dados e mantém a estabilidade do ambiente durante a carga.

O sistema de armazenamento adotado é o ClickHouse, escolhido por seu desempenho em consultas analíticas, compressão eficiente e adequação a cenários de leitura intensiva. Além disso, o paradigma OLAP contribui para a preservação da proveniência, integridade e rastreabilidade dos dados, uma vez que, após a inserção, os registros são utilizados prioritariamente para consulta e análise, sem modificações destrutivas. Por fim, os dados persistidos podem ser acessados por meio de consultas SQL, permitindo a recuperação de cromatogramas e espectros por filtros de amostra, *scan filters*, intervalos de massa e tempo de retenção. A partir dessas consultas, são geradas visualizações que reproduzem computacionalmente os resultados dos arquivos brutos, demonstrando que o método não apenas automatiza a ingestão, mas também estabelece uma base sólida para exploração analítica independente de softwares proprietários.

5. Resultados e discussões

A avaliação do método AnalyticalChemistryLake foi conduzida por meio da comparação entre três soluções de armazenamento analítico: DuckDB, PostgreSQL com TimescaleDB e ClickHouse. Os critérios considerados incluíram desempenho em consultas analíticas, eficiência de armazenamento, segurança, suporte a múltiplas requisições simultâneas, possibilidade de alteração dos dados e adequação ao paradigma OLAP. Os testes foram realizados sobre conjuntos de dados provenientes das técnicas LCHR-MS e GC-MS em modo SIM, utilizadas como cenários experimentais para validar o método.

O DuckDB foi inicialmente considerado por sua facilidade de uso e bom desempenho em análises locais. Contudo, sua adoção foi descartada por limitações relevantes ao contexto deste trabalho: ausência de mecanismos robustos de segurança e controle de acesso em ambiente compartilhado, uso de arquivos abertos (por exemplo, Parquet) e inadequação para ingestão concorrente (não projetado para múltiplas requisições simultâneas de escrita). Em termos de desempenho, observou-se eficiência inferior em consultas analíticas quando comparado ao ClickHouse, além de menor eficiência de armazenamento. Essas características inviabilizam seu uso como repositório analítico centralizado.

O PostgreSQL com TimescaleDB apresentou maior robustez transacional e melhor suporte a ambientes multiusuário. Entretanto, nos testes realizados, verificou-se desempenho significativamente inferior em consultas sobre grandes volumes de dados. Em um cenário de geração de aproximadamente 500 cromatogramas sobrepostos, o tempo de resposta variou entre 3 e 5 minutos, enquanto no ClickHouse a mesma operação foi executada em 20 a 40 segundos. Ademais, por permitir alteração dos dados armazenados, o PostgreSQL/TimescaleDB não atende plenamente ao requisito de imutabilidade desejado para um repositório analítico voltado à preservação de proveniência. Em termos de armazenamento, apresentou desempenho equivalente ou levemente inferior ao ClickHouse. A Tabela 2 sintetiza a comparação entre as soluções avaliadas.

Tabela 2. Comparação entre bancos de dados testados.

	Acurácia	**Alteração dos dados**	**Desempenho em consutas**	**Armazenamento**
DuckDB	Baixa / sem suporte a escrita concorrente	Possível	Inferior ao ClickHouse	Inferior
PostgreSQL + TimescaleDB	Alta	Possível	Lento (3 – 5 minutos)	Similiar ou inferior
Clickhouse	Alta	Não (OLAP)	Alto (20 – 40 segundos)	Superior

O ClickHouse mostrou-se a solução mais adequada ao método proposto. Por ser um banco nativamente orientado a OLAP, com armazenamento colunar e compressão eficiente, oferece alto desempenho em consultas analíticas e se ajusta ao padrão de leitura intensiva requerido pelo AnalyticalChemistryLake. A política de dados imutáveis após inserção contribui para a integridade, rastreabilidade e proveniência das informações. Do ponto de vista operacional, a implementação foi direta e o modelo tabular facilitou a adaptação da modelagem proposta, consolidando o ClickHouse como a tecnologia adotada.

Quanto ao desempenho do *workflow* por amostra, os testes indicaram tempos médios de aproximadamente 1 minuto por amostra para LCHR-MS e cerca de 10 segundos por amostra para GC-MS (SIM). Essa diferença decorre do volume de dados e do custo de conversão para mzML: para LCHR-MS, a conversão consome em média 55 segundos, enquanto as demais etapas (extração, estruturação e carga) demandam 3–4 segundos; para GC-MS, a conversão leva cerca de 7 segundos, e o restante do fluxo 2–3

segundos. As amostras de LCHR-MS continham, em média, 14 cromatogramas MS/MS e 88 cromatogramas de *single ion monitoring*, ao passo que as de GC-MS apresentavam 11 cromatogramas de *single ion monitoring*, justificando a diferença observada.

Além do desempenho computacional, observou-se ganho expressivo em eficiência operacional e armazenamento. No procedimento tradicional, a recuperação de uma amostra a partir de arquivos brutos armazenados em armários físicos ou sistemas de *backup* demanda, em média, 30–40 minutos e requer presença no laboratório. No sistema proposto, o acesso por meio de consultas previamente definidas ocorre em 5–10 segundos. Em termos de armazenamento, enquanto os arquivos brutos variam de 20 MB a 2 GB por amostra, a representação estruturada no ClickHouse ocupou, em média, ~3,4 MB por amostra (cenário com 200 amostras por técnica), evidenciando redução substancial de espaço e viabilização de análises em larga escala.

A validação dos dados armazenados foi realizada por comparação visual entre cromatogramas gerados a partir do repositório analítico e aqueles obtidos diretamente no software proprietário a partir dos arquivos brutos. A Figura 3 apresenta quatro cromatogramas: dois de LCHR-MS (um gerado via banco e outro via software proprietário) e dois de GC-MS (SIM) sob a mesma lógica. A sobreposição visual e a equivalência de perfis confirmam a fidelidade da extração, estruturação e persistência dos dados no sistema, demonstrando que o método preserva as características essenciais dos sinais analíticos.

Figura 3. Validação visual dos cromatogramas (banco x software proprietário).

Por fim, o método evidenciou vantagens em reprodutibilidade e automação. Uma vez estruturados e armazenados, os dados podem ser consultados repetidamente por SQL, permitindo a reprodução de cromatogramas e análises sem dependência de softwares proprietários ou de procedimentos manuais. Isso facilita auditorias, validações e novas "corridas" analíticas sobre dados históricos, consolidando a adoção de uma arquitetura OLAP baseada em ClickHouse como tecnicamente adequada ao método proposto.

6. Conclusão

Este trabalho apresentou o AnalyticalChemistryLake, um método de engenharia de dados para acesso, estruturação e armazenamento analítico de dados cromatográficos, implementado como um *workflow* automatizado em Python. A abordagem integra conversão para formato aberto, extração de metadados, modelagem relacional e persistência em um repositório analítico, com foco em padronização, automação e independência de *software* proprietário. Os resultados obtidos demonstram que o método se mostrou útil e eficaz para organizar, consultar e reutilizar grandes volumes de dados analíticos.

A avaliação das tecnologias de armazenamento indicou o ClickHouse como a solução mais adequada ao método proposto, superando DuckDB e PostgreSQL/TimescaleDB em desempenho de consultas analíticas, eficiência de armazenamento e alinhamento ao paradigma OLAP. A política de dados imutáveis após inserção contribuiu para a proveniência, integridade e rastreabilidade das informações, requisitos essenciais para ambientes laboratoriais e para a validação de resultados ao longo do tempo.

No aspecto operacional, o método apresentou tempos médios de processamento de aproximadamente 1 minuto por amostra para LCHR-MS e 10 segundos por amostra para GC-MS em modo SIM, com a maior parcela do tempo associada à conversão para mzML. Observou-se ainda uma redução expressiva de armazenamento, com cerca de ~3,4 MB por amostra no repositório analítico frente a arquivos brutos de 20 MB a 2 GB, evidenciando ganhos de eficiência e viabilidade para coleções de dados em larga escala.

A validação por comparação visual entre cromatogramas gerados a partir do banco e aqueles obtidos no software proprietário confirmou a fidelidade da extração, estruturação e persistência dos dados. Além disso, o acesso por SQL viabiliza reprodutibilidade, automação e exploração retrospectiva sem dependência de ferramentas proprietárias, reduzindo significativamente o tempo de acesso e ampliando o potencial de auditoria e análise.

Conclui-se, portanto, que o AnalyticalChemistryLake constitui uma solução robusta, reprodutível e independente de fabricante para a gestão de dados analíticos, particularmente adequada a cenários de consultas analíticas intensivas. Como perspectivas futuras, destacam-se a ampliação do suporte a outros formatos compatíveis com mzML e a integração com camadas avançadas de análise e visualização, de modo a expandir o alcance do método em aplicações científicas e institucionais.

Referências

François, A. (2019). pymsfilereader: Thermo MSFileReader Python bindings. GitHub. Disponível em: https://github.com/frallain/pymsfilereader.

Gaida, S., & Neumann, S. (2016). MetHouse: Raw and Preprocessed Mass Spectrometry Data. Journal of Integrative Bioinformatics.

Gleaves, J., et al. (2021). Doping prevalence in competitive sport: evidence synthesis with "best practice" recommendations and reporting guidelines from the WADA Working Group on Doping Prevalence. Sports Medicine, 51(9), 1909-1934.

Gowd, B. P., Jayasree, K., & Hegde, M. N. (2018). Comparison of artificial neural networks and fuzzy logic approaches for crack detection in a beam like structure. Int. J. Artif. Intell. Appl, 9(1), 35-51.

Inmon, W. H. (2016). Building the Data Warehouse. Robert Elliott.

Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition. Wiley.

Mardal, J., et al. (2023). Scalable Analysis of Untargeted LC-HRMS Data by Means of SQL Database Archiving. Analytical Chemistry.

Mattoso, M., et al. (2010). Towards supporting the life cycle of large scale scientific experiments. Int. J. of Business Process Integration and Management, 5(1), 79-92.

Miloslavskaya, N., & Tolstoy, A. (2016). Big Data, Fast Data and Data Lake Concepts. Procedia Computer Science.

Mogollon, N. G., et al. (2014). State of the art two-dimensional liquid chromatography: fundamental concepts, instrumentation, and applications. Química Nova, 37, 1680-1691.

Ryoo, H., et al. (2024). Identification of doping suspicions through artificial intelligence-powered analysis on athlete's performance passport in female weightlifting. Frontiers in Physiology, 15, 1344340.

Yao, Y., et al. (2015). Analysis of Metabolomics Datasets with High-Performance Computing and Metabolite Atlases. Metabolites.