

CEFET/RJ

Eidolon: Uma Abordagem para Publicação Reprodutível de Coleções Científicas de Dados

Lucas Calmon · Marcus Junior · Lucas Santos · Fernando Alexandrino
Rafaelli Coutinho · Gustavo Guedes · Diego Carvalho · Eduardo Ogasawara

Desafios na Distribuição e Reprodutibilidade de Coleções Científicas de Dados

Grandes coleções de dados científicos usadas em benchmarks precisam ser **reutilizáveis**, mas distribuí-las integralmente é impraticável. Um dataset de 800 GB pode estar bem armazenado online e ainda assim ser difícil de integrar diretamente a um experimento reproduzível.

Curados

Dados limpos, padronizados e prontos para uso em experimentos científicos.

Integrados

Compatíveis com o ambiente computacional do pesquisador, sem fricção de acesso.

Documentados

Metadados estruturados que descrevem origem, formato e forma de uso.

Leves para distribuir

Pequenos o suficiente para viajar junto com o pacote de software, sem overhead.

Fonte: artigo "Eidolon: An Architecture for the Reproducible Publication of Scientific Data Collections".

Abordagens Extremas para Distribuição de Dados Científicos: Limitações e Compromissos

Antes do Eidolon, os pesquisadores oscilavam entre duas abordagens opostas, cada uma com compromissos significativos. O Eidolon surge como uma **ponte pragmática** entre esses dois extremos.

ABORDAGEM A

Tudo fora do pacote

Os dados ficam em repositórios ou portais externos, separados do código de análise.

- ✓ Bom para armazenar grandes volumes
- □ Uso local fica fragmentado e difícil de reproduzir

ABORDAGEM B

Tudo dentro do pacote

Os dados completos são embutidos diretamente no pacote de software distribuído.

- ✓ Uso direto e imediato no ambiente local
- □ Difícil de distribuir e manter em escala

Eidolon: mantém acesso local sem carregar todo o volume de dados.

Arquitetura de Publicação de Dados Científicos: Miniatura Local e Expansão Remota Sob Demanda

> O pacote carrega uma miniatura fiel; o dataset completo fica remoto e é carregado só quando necessário.



1 — Curar e padronizar

Os dados originais são limpos, normalizados e documentados com metadados estruturados.

2 — Distribuir miniatura local

Uma representação compacta do dataset é embarcada no pacote, preservando esquema e identidade.

3 — Expandir sob demanda

Quando o experimento exige os dados completos, a função `loadfulldata()` os recupera remotamente.

A miniatura é pequena, mas preserva identidade, esquema, metadados e forma de acesso do dataset completo. | Fonte: artigo "Eidolon: An Architecture for the Reproducible Publication of Scientific Data Collections".

Fluxo Operacional de Publicação: Etapas e Mecanismos

Do dado original até o workflow experimental com expansão remota. O passo crítico é a **padronização**: nome, descrição, formato, categoria, fonte, referências e exemplo de uso precisam ficar consistentes em todos os datasets da coleção.

1 — Fontes

Arquivos CSV originais e demais fontes brutas de dados científicos.

2 — Curadoria

Seleção, limpeza e validação dos dados relevantes para a coleção.

3 — Padronização

Uniformização de nomes, formatos, categorias e metadados de cada dataset.

4 — Artefato completo

Geração do dataset estruturado com listas, onde cada lista é um CSV temático.

5 — Pacote local

Empacotamento do artefato completo junto ao código do pacote de software.

6 — Miniatura local

Geração da versão compacta embarcada no pacote para distribuição leve.

7 — Expansão

Recuperação remota do dataset completo via `loadfulldata()` quando necessário.

8 — Workflow

Uso do dataset completo ou miniatura no experimento reproduzível do pesquisador.

Documentação Padronizada de Datasets: o Caso `climate`

Cada dataset da coleção é documentado diretamente no código R usando o padrão roxygen2. O bloco abaixo é o registro real do dataset `climate` em `tspredbench.R` — todos os campos são obrigatórios e seguem a mesma estrutura para todos os datasets da coleção.

```
#' @name climate
#' @title FAOSTAT Temperature Change on Land
#' @description
#' Statistics of surface temperature anomalies on land, based on NASA-GISS GISTEMP data.
#' Data Type: Temperature Anomalies. Category: Environment. Creation Date 2024.
#' @docType data
...
#' @references
#' FAO, 2024. FAOSTAT Land, Inputs and Sustainability; ...
#' @source \href{https://data.giss.nasa.gov/gistemp/}{NASA-GISS GISTEMP}
"climate"
```

`@description`

Tipo de dado, categoria e data de criação — contexto imediato para o pesquisador sem necessidade de consultar fontes externas.

`@references`

Rastreabilidade completa até a fonte primária (FAO, NASA-GISS), com citações acadêmicas detalhadas para reprodutibilidade.

`@examples`

Código executável que demonstra carregamento e visualização da série, incluindo o uso opcional de `loadfulldata()`.

Fonte: github.com/cefet-rj-dal/tspredit/blob/main/R/tspredbench.R

Validação empírica: os pacotes tspredit e tspredbench

A arquitetura foi testada com datasets de **previsão de séries temporais**, reunindo 17 coleções curadas em domínios variados. A função-chave `loadfulldata()` transforma a miniatura local no objeto completo sem alterar a forma de programar.

8

Benchmarks

EUNITE, NN3, NN5, CATS, SantaFe e outros conjuntos clássicos de competição.

2

Macroeconomia

ipeadata.d e ipeadata.m — séries do Instituto de Pesquisa Econômica Aplicada.

6

Multidomínio

bioenergy, climate, emissions, fertilizers, gdp e pesticides — dados globais da FAO.

1

Mercado financeiro

stocks — série histórica de preços de ações para análise preditiva.

Aplicação Prática: Acesso e Visualização do Dataset Climate

O fluxo completo — do carregamento à visualização — usando o dataset climate (FAOSTAT/NASA-GISS). A chamada `loadfulldata()` recupera o dataset completo remotamente a partir da miniatura local.

Passo 1 — Carregar biblioteca e dados

```
library(tspredit)
data(climate)
climate <- loadfulldata(climate)
```

Passo 2 — Inspecionar número de séries

```
cat("Series available:", length(climate), "\n")
```

```
## Series available: 10
```

Passo 3 — Visualizar chaves disponíveis

```
head(names(climate))
```

```
## [1] "usa_climate"     "china_climate"   "germany_climate"  "japan_climate"
"india_climate"   "uk_climate"
```

Passo 4 — Inspecionar valores da primeira série

```
head(climate[[1]])
```

```
##   1961   1962   1963   1964   1965   1966
##  0.003 -0.027  0.425 -0.281 -0.360 -0.212
```

Passo 5 — Plotar a série temporal

```
ts.plot(climate[[1]], ylab = "Temperature change", xlab = "Year", main = names(climate)[1])
```

Redução de Volume sem Comprometimento da Usabilidade

No estudo de caso com séries temporais, a redução de tamanho entre o dataset completo e a miniatura distribuída no pacote foi expressiva — demonstrando que a arquitetura **não compromete a usabilidade** ao ganhar leveza.

152,5MiB

tspredbench completo

Volume total do dataset remoto com todos os dados de séries temporais curadas.

64,6KiB

data/ em tspredit

Tamanho da miniatura embarcada no pacote local, pronta para uso imediato.

2416×

Redução de tamanho

Fator de compressão entre o artefato completo e a miniatura distribuída no pacote.

A arquitetura permite ao usuário descobrir, testar e entender os dados localmente — e só busca o volume completo quando o experimento realmente exige.

Avaliação Crítica: Vantagens e Restrições da Arquitetura

A proposta é robusta para coleções grandes demais para embutir no pacote, mas que precisam ficar próximas do ambiente de análise. Ainda assim, a maturidade da arquitetura depende de **manutenção ativa e infraestrutura remota confiável**.

Benefícios

- Pacote significativamente mais leve, sem abrir mão do acesso aos dados completos.
- Descoberta e inspeção local dos dados antes de qualquer download pesado.
- Mesma interface de programação para miniatura e dataset completo.
- Mais próximo do ideal de workflow reprodutível em ciência de dados.

Limitações

- Miniaturas e dados completos precisam ficar sincronizados ao longo do tempo.
- Depende de hospedagem remota estável para a expansão funcionar corretamente.
- Ainda falta suporte a checksum e versionamento fixo dos artefatos remotos.
- Necessita ser testado em outros ecossistemas além do R para validação ampla.

Considerações Finais

> O Eidolon não é apenas "guardar dados fora". É criar um **contrato confiável** entre a miniatura local e o dataset completo.

A arquitetura torna grandes coleções científicas mais fáceis de publicar, descobrir e reutilizar, sem sacrificar a leveza do pacote nem a fidelidade dos dados. Ao separar a representação local da expansão remota, o Eidolon aproxima a publicação de dados do ideal de ciência aberta e reprodutível.

Publicar

Coleções grandes ganham um fluxo padronizado de curadoria e distribuição.

Descobrir

Pesquisadores exploram os dados localmente antes de qualquer download pesado.

Reutilizar

A mesma interface serve para miniatura e dataset completo, sem reescrever código.

Obrigado!



CEFET/RJ

Agradecimentos

FAPERJ

CAPES

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

Lucas Calmon · Marcus Junior · Lucas Santos · Fernando Alexandrino · Rafaelli Coutinho · Gustavo Guedes · Diego Carvalho · Eduardo Ogasawara

https://eic.cefet-rj.br/dal