Tutorial do GE.db

Sobre o sistema

Configuracoes iniciais

Customizacao dos dados mestres

Workflow

Levantar o processo em ambiente HPC

Processo de ETL (workfloes e dataflow)

Descricao dos processos

Interacoes com as tabelas de dados mestres

# Sobre o Sistema

O sistema IGEM foi desenvolvido como uma plataforma para hospedar APP que poderão compartilhar recursos e interagir entre si com uma única base de dados. O primeiro APP criado foi o GE (Gene x Exposome) com o objetivo de coletar bases de dados externas, identificar termos como genes e exposomes e criar uma base de conhecimento. Essa informação sera armazeda para consultas dinâmicas e exploratórias posteriormente.

O APP possui controle de dados mestres, mecanismo de NLP para identificação de termos e bigdata para mapeamento e redução dos termos correlacionados.

Configuracoes Iniciais

O sistema IGEM possui uma arquitetura modular divida em dois grupos, sendo um de componentes que hospeda serviços como gerenciamento do File System, Banco de dados e Interfaces WEB e outro grupo de APPs, sendo desenvolvido ate o momento o APP GE.

O app GE hospeda processos e ate o momento teremos:

Data WorkFlow

COLLECTOR

PREPARE

COMMUTE

MAPREDUCE

Master Data Load

LOADER

Manutencao

Maintenance

Runserver

Createsuperuser

Makemigrations

Migrate

Filter e Consultas

Filter

Ainda não temos um pacote de instalação, sendo necessário apenas copiar os arquivos fontes para um diretório e a pasta com o ambiente virtual (no ambiente ICDS/HPC foi criada um Conda para utilizar a versão 3.10.4 do Python)

O IGEM esta disponível no ICDS ambiente em /gpfs/group/mah546/default/sw/igem, onde teremos três sub pastas:

/igem\_db : hospeda a estrutura do banco de dados POSTGRES e a base do sistema em igemdb

/igem\_env : hospeda os arquivos do ambiente virtual conda

/src : hospeda o sistema e pastas de apoio.

Dentro do diretório src, teremos:

/ge/ : todo os códigos fontes e interfaces para o funcionamento do APP GE

/loader/ : todos os arquivos de entrada para carga de dados mestres e diretório de saída do processo FILTER

/psa/ : Persiste Store Area para armezenar os arquivos de bases de dados baixados e em processamento pelo processo de ETL. Cada DATASET terea sua própria subpasta dentro da PSA.

/src/ : hospeda os códigos fontes dos componentes, configurações e parametrizações do IGEM

/templantes/: hospeda as interfaces web comum no IGEM

O sistema IGEM é modular quanto a escola do banco de dados, para o projeto no ICDS, a escolha foi pelo POSTGRES que possui melhores características para a necessidade do GE.db, permitindo melhor desempenho e segurança no ambientes de HPC. O único problema ė que necesse ambiente ele tem comportamento simple user, sendo necessário instanciar sempre antes de seu uso. Porem em ambiente de cloud como serviço, ele entrega melhor performance e multi-users comparado com SQLite

A configuração do banco de dados está no arquivo fonte settings.py em :

Graphical user interface, text, application

Description automatically generated

# Customizacao dos Dados Mestres

Os dados mestres possuem uma função primordial para o funcionamento do sistema, sendo eles responsáveis por direcionar as origem dos dados e pela filtragem e ligação dos termos para futura consultas.

Antes de iniciar a coleta de dados, sera necessária a parametrizações dos Dados Mestres que poderá ser via lote ou individual em interface web.

Os dados mestres poderão ser configurados na seguinte ordem:

- DATABASE

- KEYGE-PREFIX

- DATASET

- KEYGE-GROUP

- KEYGE-CATEGORY

- KEYGE

- KEYGE-WORD

A seguir iremos detalhar cada dado mestre

DATABASE

Um registro de DATABASE aponta para um determinado conjunto de dados externos, possuindo informações de autenticações, categoria dos dados hospedados e descrição. Ela sera referenciada nos DATASET (próximo dado mestre).

O acesso via web sera em <http://127.0.0.1:8000/admin/ge/database/>

Importacao de dados em lote sera através do comando:

$ Python manage.py loader --database.

O arquivo para importacao em lote devera ter os campos:

Database

Descricao

Website

Category

O arquivo devera ser salvo em ../igem/src/loader/database.csv antes de executar o comando de importação em lote.

A importação em lote ira desconsiderar todos os registros repetidos ou já existentes na GE.db

DATASET

Um registro de DATASET armazena todas informações necessárias para a extração externa e controla as ligações de termos relacionados pelo processo de mapeamente. O controle de extração de dados externos ocorre por DATASET. Cada DATASET será composto pelos campos:

DATABASE: Agrupamento de DATASET, controlando autenticao de acesso.

DATASET: Abreviacao para identificação do DATASET

Descricao: Breve descrição sobre o objetivo do DATASET

Ativado: Flag informando se o DATASET está ativo e será considerado no processo de ETL

Group Attributes: Campos para controle de caminho de extração e tipo de arquivo

Source path from Internet: Flag para rotiar o caminha de extracao, se ativo sera via http e desativado ira considerar como caminho de arquivo local.

Source path: Caminho em que o arquivo do DATASET esta hospedado

Source file name: nome do arquivo com os dados originais

Source file format: formato do arquivo com os dados originais. Essa informação será importando para tratamento de conversão para o formato de ingestão dos dados no processo ETL. Caso compactado, informar apenas o formato da compactação, tipo ZIP, GZ..

Source file sep: informar o tipo de separador de arquivo caso tenha. Para separador tabular, utilizar /n

Source file skiprow: Informar quantidade de linhas que serão eliminadas no processo ETL, muitos arquivos possuem em suas primeiras linhas informações estruturais que não são necessários no processo de ETL.

Source compact: Flag para controlar se o arquivo esta compactado, se não tiver marcado, considera-se como arquivo não compactado.

Target file name: Nome do arquivo após descompactação

Target file format: Formato do arquivo após descompactação, esse campo sera o formato real do arquivo, tipo CSV, TXT

Keep file: Flag se selecionada ira manter o arquivo após tratamento de dados para futura consulta. Importante analisar o consumo de espaço de armazenamento, pois manter os arquivos poderá consumir espaço desnecessário. Novas atualizações subtituira os arquivos existentes.

Group Columns: Controla regras para tratamento dos dados extraídos para formato compatível com o sistema GE.db. Considerar apenas as colunas com códigos padrão MEsH NIH.

Column Sequence: Numero da coluna que recebe a regra

Column Name: Nome da coluna para orientar e ajudar na identificação da regra aplicada

Active: flag que informa se a regra esta ativa

PREFIX: informar o prefixo do termo que sera considerado e adicionado a informação da coluna.

O acesso via web sera em <http://127.0.0.1:8000/admin/ge/dataset/>

Importacao de dados em lote sera através do comando:

$ Python manage.py loader –dataset

Na importacao em lote nao serao considerados os parâmetros do Group Columns de transformação. Após a importação em lote, será necessário acessar cada DATASET via web e parametrizar as regras de colunas caso existam.

O arquivo devera ser salvo em ../igem/src/loader/dataset.csv antes de executar o comando de importação em lote.

GROUP

Os grupos do sistema irão auxiliar no tipo de termos de pesquisas e serão atribuídos no KEYGE explicado mais a frente.

Possui apenas dois campos: Group e descrição.

O acesso via web sera em <http://127.0.0.1:8000/admin/ge/group/>

Importacao de dados em lote sera através do comando:

$ Python manage.py loader –group

O arquivo devera ser salvo em ../igem/src/loader/group.csv antes de executar o comando de importação em lote.

Para o nosso objetivo, teremos dois grupos, environment e genomic

CATEGORIA

As categoria auxiliam na identificação e pesquisa dos termos e serão atribuídas no KEYGE explicado na sequencia.

Possui apenas dois campos: Categoria e descrição.

Importacao de dados em lote sera através do comando:

$ Python manage.py loader –category

O arquivo devera ser salvo em ../igem/src/loader/category.csv antes de executar o comando de importação em lote.

KEYGE-PREFIX:

Grupos para identificação do tipo de informação por coluna dos dados extraídos. Essa informação sera utilizada no momento da parametrização do DATASET no grupo de Colunas.

KEYGE

Cadastro dos termos que serão pesquisados ou convertidos no processo de ETL de cada DATASET. A conversão ira ocorrer com apoio do cadastro KEYGE-WORD explicado a seguir.

O KEYGE será o ponto central no momento das pesquisas e o resultado do ETL sera a ligação de duas KEYGE.

O KEYGE possui os campos:

KEYGE: termo de pesquisa

Descricao: descrição do termo e auxiliar em pesquisas abertas e exploratórias

Group: Agrupamento dos termos para auxiliar nas pesquisas e agrupamento

Category: Agrupamento dos termos em um segundo nível para auxiliar nas pesquisas e agrupamentos.

O acesso via web sera em <http://127.0.0.1:8000/admin/ge/keyge/>

Importacao de dados em lote sera através do comando:

$ Python manage.py loader –keyge

O arquivo devera ser salvo em ../igem/src/loader/keyge.csv antes de executar o comando de importação em lote.

No arquivo informar os nomes de GROUP e CATEGORY. O processo identificará a chave automaticamente. Caso os valores desses dois campos não estiverem cadastrados, sera retornado uma mensagem de erro e não sera realizada a carga. Proceda com o cadastro das informações faltantes antes de nova tentativa de carregar o KEYGE em lote.

KEYGE-WORD

Cadastro das palavras que serão convertidas para os termos KEYGE. O sistema permite cadastrar palavras compostas.

Teremos os seguintes campos:

KEYGE: Sera o termo para conversão

WORD: palavra ou conjunto de palavras que será convertido para o KEYGE

ACTIVE: Flag para ativar o relacionamento

COMMUTE: Flag utilizado para converter, caso seja o mesmo critério entre o KEYGE e WORD, desative esse flag para reduzir consumo de memoria durante o processo de ETL

O acesso via web sera em <http://127.0.0.1:8000/admin/ge/keyword/>

Importacao de dados em lote sera através do comando:

$ Python manage.py loader –keyword

O arquivo devera ser salvo em ../igem/src/loader/keyword.csv antes de executar o comando de importação em lote.

**Processo de ETL**

O processo de ETL (Extracao, Transformacao e Loader) sera a etapa em que o sistema ira buscar os dados de fontes externas, tratar esses dados para um padrão compatível, buscar as relações de termos e gravar na GE.db. Todo o processo ocorre com base nos parâmetros e dados mestres configurados breviamente, por isso sera de extrema importância a atenção nas parametrizações e cadastros dos dados mestres.

Para melhor gestão de recursos durante a fase de ETL, o workflow do processo foi divido em 5 fases distintas:

1 fase – COLLECT: Processo responsável por selecionar os DATASETS ativos e para cada um verificar se há disponível nova versão dos dados, caso sim, ira extrair os novos dados e se necessário, descompactar o arquivo extraído, deixando o arquivo em uma extrutura PSA (persiste Storage Area).

O processo também realiza a atualização dos logs e controles de versões.

A execução do processo ocorre por linha de comandos, permitindo o desenvolvimento de script para execução em serie com os demais fases do processo. Sempre executar os comando dentro da pasta src do programa. A versão de execução das fase em interface web ainda esta em desenvolvimento.

$ python manage.py collect --run {all or dataset}

Inicia a coleta dos novos conjunto de dados, baixando da internet ou de outro local os novos arquivos de dados e disponibilizando na pasta PSA para as próximas fases. Caso informado a opção all, o processo ira executar para todas as DATASET ativas na tabela de dados mestres, ou informar a DATASET especifica para execução isolada.

$ python manage.py collect --reset {all or dataset}

A opcao reset ira eliminar todo o controle de fases, inclusive ira eliminar o controle da ultima versão processada, permitindo reprocessar DATASET novamente. Caso informe a opção all, todas as DATASET ativas serão reiniciadas ou informar apenas uma DATASET para reset.

$ python manage.py collect –show

A opcao show ira mostrar todas as DATASET do cadastro e seus status.

$ python manage.py collect --activate {dataset}

A opcao activate ira ativart uma DATASET especifica, permitindo a extração dos dados.

$ python manage.py collect --deactivate {dataset}

A opcao deactivate ira desativar uma DATASET especifica, inibindo que ela faça parte dos arquivos baixados futuros.

2 fase – PREPARE: Nessa segunda fase do processo tem o objetivo de transformar os dados originais, assim reduzindo a necessidade de recursos computacionais nas próximas fases. Com base nos parâmetros do DATASET configurados breviamente, nessa fase teremos:

Eliminacao de linhas de cabeçalho

Delecao de columnas desnecessárias

Transformacao de ID Columns com indentificadores sufixos

Replacement os terms

Eliminacao do arquivo original.

A saída sera um novo arquivo temporário para consumo na próxima fase.

# python manage.py prepare --run {all or dataset}

Ira iniciar a fase de preparação dos dados para todos os DATASET ou apenas uma especificada. Importante ter o arquivo na PSA, caso contrario o sistema ira exibir um warning.

$ python manage.py prepare --reset {all or dataset}

A opcao reset ira reiniciar o controle para todas ou uma especifica DATASET a fase de preparação e as duas posteriores.

3 fase – Commute: Nessa fase o processo ira substituir palavras pelos termos (keyge) com base no cadastro Keyword previamente parametrizado.

# python manage.py commute --run {all or dataset}

Ira iniciar a fase de comutação de termos dos dados para todos os DATASET ou apenas uma especificada. Importante ter o arquivo na PSA, caso contrario o sistema ira exibir um warning.

$ python manage.py commute --reset {all or dataset}

A opcao reset ira reiniciar o controle para todas ou uma especifica DATASET a fase de comutacao e da próxima fase

4 fase – MapReduce: Ultima etapa do processo. Possui mecanismo para encontrar Keyges (termos) por linha chamado de Mapper e em seguida ativar o subprocesso Reducer que ira contar a quantidade de ligações encontradas no conjunto de dados. Após todo o processamento, o resultado será gravado na tabela Keylinks. Importante ressaltar os novos dados irão substituir integralmente os dados anterios do dataset em processamento.

# python manage.py commute --run {all or dataset}

Ira iniciar a fase de MapReduce de termos dos dados para todos os DataSET ou apenas uma especifica. Importante ter o arquivo na PSA, caso contrario o sistema ira exibir um warning. Essa fase ocorre grande consumo de memoria e processamento, por isso sera importante alocar recursos compatíveis com o tamanha dos dados processados.

$ python manage.py commute --reset {all or dataset}

A opcao reset ira reiniciar o controle para todas ou uma especifica DATASET na fase atual.

Em todos os comandos com argumento run possuel multiprocessamento e controle para chunk de arquivos. Porem sera necessário o balanceamento entre tamanho dos arquivos extraídos e os recursos alocados como memoria e quantidade de processadores.

Monitor do Workflow

Alem dos comandos reset demosntrados acima para controle do workflow do processo, o sistema também possui uma interface web em que o usurio poderá consultar e gerenciar o fluxo e status das fases.

Navege <http://127.0.0.1:8000/admin/ge/wfcontrol/> ou selecione no aplicativo GE a opção Dataset – Workflow

Na primeira serão exibidos os DATASET que já foram iniciados, com as seguintes colunas:

DS STATUS: informa se o DATASET esta ativo ou não para processamento das 4 fases

DATASET: Abreviacao da DATASET

LAst update dataset: Data da ultima atualização dos dados.

Source File Version: Versao do ultimo arquivo processado.

As próximas 4 columnas exibem os status dos processamentos por fase, o símbolo verde exibe status finalizado com sucesso e o vermelho ainda não processado.

Os dados somente estavao disponíveis na base GE.be após a execução de todas as fases com sucesso.

A screenshot of a computer

Description automatically generated with medium confidence

Ao selecionar um DATASET, sera aberta a tela de detalhes permitindo realizar modificações especificas por campo.

Graphical user interface, text, application

Description automatically generated

Importante: não sera necessário incluir novos DATASET no monitor do Workflow. O sistema automaticamente ira criar um novo controle após a primeira carga de dados do DATASET. Caso um dos registros de workflow for eliminado, ele também sera criado após a próxima carga de dados.

# Manutencao dos dados

Para auxiliar na gestão e manutenções especificas dos dados já armazenados no banco de dados, esta em desenvolvimento um processo em linha de comando para realizar as principais atividades.

Já esta disponível a primeira função de eliminação completa de tabelas:

$ python manage.py maintenance –delete {table}

# Consulta de dados

O modulo GE.filter esta em desenvolvimento, porem esta disponível para acesso rápido o comando filter quer ira extrair o conteúdo de determida tabela e gravar em um arquivo de saída no diretório /loader/out\_from\_filter.csv

$ python manage.py filter –{table}

Table disponiveis:

* Dataset
* Database
* Keyge
* Keylink
* Group
* Category
* Keyword

Esta em desenvolvimento uma interface para filtro da tabela KEYLINK e também através de linhas de comando.

# Levantar o processo em ambiente HPC

The IGEM system hosts the GE.db and GE.filter modules and is available in the ICDS environment for everyday use.

This document aims to demonstrate how to start IGEM and operate in command line and web interface environments.

**Start the system:**

Open an ICDS session and run the following commands:

***$ module load anaconda3***

***$ source activate /gpfs/group/mah546/default/sw/igem/igem\_env***

***$ cd /gpfs/group/mah546/default/sw/igem/igem\_db***

***$ pg\_ctl -D igemdb -l logfile start***

After these commands, we will have the session running a virtual environment with Anaconda3 running Python 3.10. We also activated the Database in Postgres. To check if everything is operational:

Check if you are running Python 3.10.4​​​​​​​

***$ Python -V***

Check if the database services are running

***$ ps aux | grep postgres***

IMPORTANT: once the Database services start, it will no longer be required to initialize again if you use the system in different ICDS sessions. However, due to the HPC architecture, it will only be possible for a user or job to activate the services. Therefore, after used the system, disable the database services as follows:

***$ ps aux | grep postgres***

*Graphical user interface, text

Description automatically generated*

*Disable the service that points to the base path (the case above: ID 5775)*

***$ kill {id}***

**Command Line**

The IGEM system has a master program that performs all operations triggering processes and parameters.

To run the program + process, we will go to the folder:

***$ cd /gpfs/group/mah546/default/sw/igem/src***

once in the folder we will type:

***$ python manage.py {process} (parameters}***

We have so far developed the process:

* loader
* collect
* prepare
* switch
* mapreduce
* createsuperuser
* runserver

Example of a complete command:

***$ python manage.py collect --show***

***$ python manage.py collect --run all***

***$ python manage.py collect --reset all***

The runserver process will be a session that will only run in an interactive RHEL7 session explained in the next topic

**Interface Web**

In the ICS-ACI Portal create a session in the RHEL7 interactive Desktop:

After the session is available, access it via Launch noCNV in the New tab button:

Graphical user interface, text, application

Description automatically generated

Go to the Mate Terminal:

A screenshot of a computer

Description automatically generated

Load the virtual environment:

***$ module load anaconda3***

***$ source activate /gpfs/group/mah546/default/sw/igem/igem\_env***

***$ cd /gpfs/group/mah546/default/sw/igem/src***

If the database has not started previously, you can check with the command:

***$ ps aux | grep postgres***

            And start:

***$ cd /gpfs/group/mah546/default/sw/igem/igem\_db***

***$ pg\_ctl -D igemdb -l logfile start***

With the conda virtual environment and database running, go to the src folder and run:

***# python manage.py runserver***

Text

Description automatically generated

Go to the browser (firefox) and access <http://127.0.0.1:8000/admin>

            Username: igem / Password: igem

Graphical user interface, text, application, email

Description automatically generated

Within IGEM's administrative interface, we can navigate through the system's tables and perform operations such as adding new datasets, new keyge, mapping with words, or even consulting the processed links.

Graphical user interface

Description automatically generated

Important: Extracting, transforming, and loading the external data to the GE base through JOBs in the ICS-b environment using multiprocessing.

After used the system, disable the database services as follows:

***$ ps aux | grep postgres --color=auto postgres***

A screen shot of a computer

Description automatically generated with low confidence

*Disable the service that points to the base path (the case above: ID 5775)*

***$ kill {id}***

***$ conda deactivate***