



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

## *CoExpr*

Identificação de grupos de co-expressão entre genes de  
proteínas mitocondriais

Mestrado Integrado em Engenharia Informática e Computação

**Paradigmas de Programação**  
PRODEI018

Henrique Manuel Martins Ferrolho  
[201202772 - ei12079@fe.up.pt](mailto:201202772-ei12079@fe.up.pt)

João Alberto Trigo de Bordalo Morais  
[201208217 - ei12040@fe.up.pt](mailto:201208217-ei12040@fe.up.pt)

João Filipe Figueiredo Pereira  
[201104203 - ei12023@fe.up.pt](mailto:201104203-ei12023@fe.up.pt)

8 de Julho de 2016

## Resumo

A quantidade elevada de informação recolhida nos dias de hoje, nas mais diversas áreas científicas, constitui um problema grave de desorganização.

Mais informação não é sinónimo de informação com qualidade. É preciso desenvolver técnicas para organizar o caos de informação existente, filtrar dados desnecessários, e saber tirar conclusões relevantes que sumariem o aglomerado de dados.

A plataforma desenvolvida com este projeto visa colmatar essa falha. O grupo desenvolveu uma aplicação modular capaz de organizar, filtrar, e recolher informação útil, que é por sua vez gravada em bases de dados para que a comunidade científica possa, de uma forma muito intuitiva, explorar dados convenientes e relevantes.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Descrição do sistema</b>	<b>5</b>
2.1	Descrição conceptual . . . . .	5
2.1.1	Funcionalidades . . . . .	5
2.1.2	Estrutura do programa . . . . .	5
2.1.3	Linguagens de programação . . . . .	6
2.2	Implementação . . . . .	7
2.2.1	Detalhes da implementação . . . . .	7
2.2.2	Ambiente de desenvolvimento . . . . .	8
<b>3</b>	<b>Conclusão</b>	<b>9</b>
<b>4</b>	<b>Melhoramentos</b>	<b>10</b>
<b>5</b>	<b>Recursos</b>	<b>11</b>
5.1	Software . . . . .	11
5.2	Outros . . . . .	11
<b>A</b>	<b>Apêndice</b>	<b>12</b>
A.1	Manual do utilizador . . . . .	12
A.1.1	Execução da plataforma . . . . .	12
A.1.2	Preparação da base de dados . . . . .	13
A.2	Screenshots da plataforma . . . . .	17

# 1 Introdução

No âmbito da Unidade Curricular de Paradigmas da Programação do Programa Doutoral em Engenharia Informática foi proposto o desenvolvimento de uma plataforma de identificação de grupos de co-expressão entre todos os genes de proteínas mitocondriais.

A plataforma pode ser vista como um motor de busca a uma base de dados com diversos tecidos humanos, que contém pares de genes relevantes de cada tecido e a sua respetiva semelhança. Existem filtros para refinar os resultados de pesquisa: um para selecionar um dado intervalo de semelhança entre genes, e outro para a ordenação do coeficiente de semelhança.

A ferramenta deve por isso ser usada como um auxílio à investigação, poupando tempo aos investigadores, que podem tirar conclusões com base na informação presente na base de dados muito mais rapidamente do que se tivessem que gerir toda essa informação manualmente num sistema não especializado.

O interesse numa ferramenta desta escala não se limita meramente ao nível académico, mas também a centros de investigação, como por exemplo o Instituto de Patologia e Imunologia Molecular da Universidade do Porto - IPATIMUP.

## 2 Descrição do sistema

### 2.1 Descrição conceptual

#### 2.1.1 Funcionalidades

Este projeto processa e organiza a informação genética de tecidos humanos presente num ficheiro com 4.7 GB em bases de dados adequadas. Por sua vez, essas bases de dados podem ser acedidas por uma plataforma simples e intuitiva, para que o utilizador possa pesquisar e navegar pela informação genética de uma forma *user-friendly*.

Essa plataforma permite a consulta dos coeficientes de correlação entre todas as combinações possíveis de pares de genes de proteínas mitocondriais para todos os tecidos humanos relevantes. É possível consultar os coeficientes para um tecido individual, e ainda filtrar esses resultados com base num intervalo de correlação. Também existe a opção de ordenar os resultados por ordem crescente ou decrescente de coeficiente de correlação.

#### 2.1.2 Estrutura do programa

O projeto pode ser dividido em quatro módulos distintos, cada um com a sua própria função e contribuição para a plataforma final, que é o ponto de contacto com o utilizador final.

Na figura 1 é possível ver um diagrama dos módulos que constituem o projeto, tal como as suas inter-relações.

O primeiro módulo é o *Parser*. O *Parser* é responsável por organizar, filtrar, e separar toda a informação contida no ficheiro genético de 4.7 GB disponibilizado em [GTEx Portal](#) - simbolizado pela nuvem, no diagrama. Esse ficheiro, contém informação de diversos tecidos, valores de amostras desses tecidos, e outras informações que não são relevantes para a finalidade da plataforma.

Por estas razões, é apropriada a existência de um módulo inicial que trate e filtre a informação contida nesse ficheiro.

O segundo módulo é um *script* muito simples. O seu propósito é fazer *reset* às bases de dados do sistema, deixando-as preparadas para serem preenchidas pelo próximo módulo.

O terceiro módulo é um programa cuja tarefa é preencher as bases de dados dos tecidos, que por sua vez vão ser consultadas pela plataforma final. Os ficheiros gerados pelo módulo 1 - o *Parser* - são o ponto de partida deste módulo que, com a informação contida nesses ficheiros, calcula o coeficiente de correlação para todos os pares de genes possíveis em cada tecido, e guarda essa informação na base de dados do tecido correspondente.

Finalmente, o quarto e último módulo é a plataforma direccionada ao utilizador final. Este módulo usa as bases de dados que são produto resultante dos três módulos anteriores.

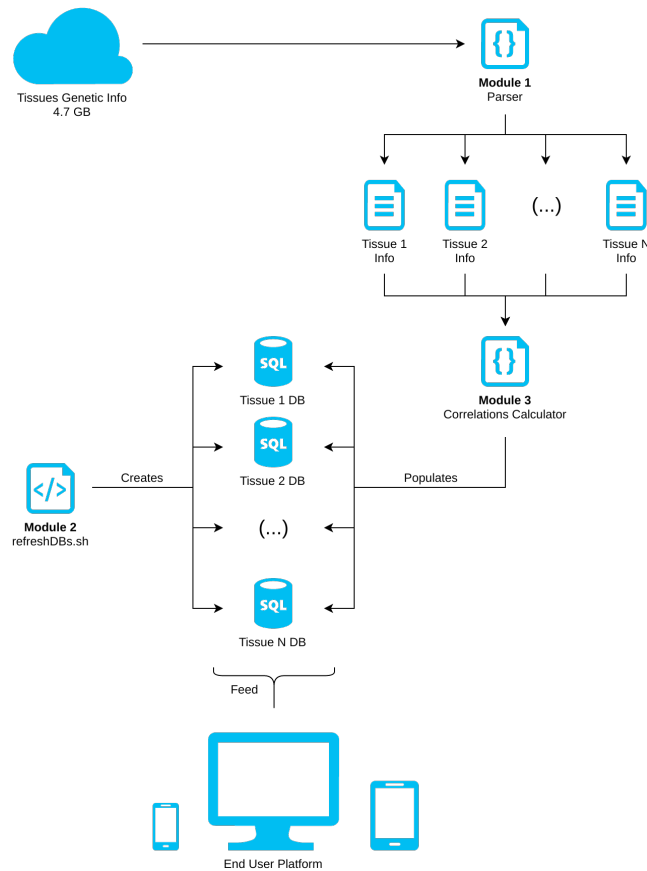


Figura 1: Diagrama dos módulos do projeto, as relações entre eles, e as interações com a informação.

### 2.1.3 Linguagens de programação

- C++, SQLite

Os módulos 1 e 3 (o *Parser* e o programa responsável por calcular os coeficientes de correlação) foram implementados em C++ 11.

O módulo 3 usa ainda SQLite para poder gravar nas bases de dados a informação gerada nos seus cálculos.

A linguagem de programação C++ enquadra-se nos paradigmas procedimental, e orientado a objetos. SQLite enquadra-se no paradigma declarativo.

- Bash

O módulo 2 (o *script* de *reset* às bases de dados) é um *Shell Script*.

Os paradigmas em que se enquadra são o imperativo, e scripting.

- PHP, JavaScript, CSS, HTML

O módulo 4 é a plataforma principal do projeto. Está implementado em Laravel, uma *framework* de PHP.

A plataforma segue um padrão MVC - *Model, View, Controller*: os controladores foram implementados em PHP; as vistas foram construídas em HTML e CSS. Foi ainda usado JavaScript, e as bibliotecas de JavaScript: AJAX e jQuery, para tornar o motor de busca mais responsivo e dinâmico.

PHP enquadra-se nos paradigmas procedimental, e orientado a objetos. JavaScript é também uma linguagem multi-paradigma: scripting, orientado a objetos (baseada em protótipos), imperativa, e funcional. HTML e CSS enquadram-se no paradigma declarativo.

## 2.2 Implementação

### 2.2.1 Detalhes da implementação

O módulo principal do projeto é a plataforma desenvolvida em Laravel. As bases de dados são uma parte fundamental da plataforma: se as bases de dados não estiverem preenchidas com os dados genéticos, a plataforma não tem qualquer utilidade.

Assim sendo, todos os restantes módulos do projeto existem com a finalidade de processar o conteúdo do ficheiro inicial de 4.7 GB obtido no Portal GTEx. Esses módulos interagem uns com os outros, e podem ser vistos como uma "linha de montagem" que acaba por preencher devidamente as bases de dados da plataforma com a informação estritamente relevante e necessária.

O código-fonte em C++ 11 do primeiro módulo encontra-se no ficheiro `parser.cpp`. Este módulo processa o ficheiro inicial obtido no Portal GTEx: o programa separa o conteúdo desse ficheiro em múltiplos ficheiros, um para cada tecido relevante (sendo que os tecidos não relevantes têm menos de 10 amostras, e são descartados).

O *script* de *reset* às bases de dados constitui o segundo módulo, e foi implementado em *Shell Script (Bash)*. O *script* tem duas fases: uma primeira que apaga todos os ficheiros `.sqlite` existentes; e uma segunda que faz *touch* a todos os ficheiros `.sqlite` necessários para a plataforma.

Depois da execução deste *script*, as bases de dados encontram-se inicializadas e vazias, prontas a serem preenchidas pelo próximo módulo.

O terceiro módulo também foi implementado em C++ 11, e o seu código-fonte encontra-se no ficheiro `correlationsToDB.cpp`.

Este módulo recebe um parâmetro que corresponde ao nome do tecido cuja base de dados se pretende preencher. O programa começa por processar o ficheiro do tecido correspondente, que foi preparado pelo *Parser* (módulo 1).

De seguida, o programa calcula o coeficiente da correlação de Pearson para todas as combinações possíveis entre um gene mitocondrial e todos os outros genes do tecido em questão. Finalmente, essa informação é gravada na base de dados do respetivo tecido (base de dados essa que foi inicializada pelo módulo 2), na forma do seguinte tuplo: (*ID de correlação, gene mitocondrial, gene do tecido, coeficiente de correlação de Pearson*).

Foram ainda tomadas as seguintes medidas para otimizar e tornar mais rápida a inserção e leitura dos tuplos na base de dados:

- O modo *synchronous* do SQLite foi desativado
- O *journal\_mode* foi definido para usar a memória RAM
- Os tuplos são inseridos todos numa transação
- As inserções são feitas através de um *prepared statement*
- Todas as bases de dados têm um índice afeto à coluna da correlação para aumentar a velocidade dos *selects*

O quarto e último módulo é a plataforma, desenvolvida em Laravel (uma *framework* de PHP). O código-fonte de toda a plataforma encontra-se dentro da pasta **coexpr**.

A plataforma é um motor de busca e uma ferramenta de visualização do conteúdo das bases de dados, que é inserido pelo módulo anterior.

### 2.2.2 Ambiente de desenvolvimento

De seguida é apresentada uma descrição do ambiente de desenvolvimento, e as versões de cada ferramenta utilizada.

**Sistema Operativo:** Linux

**Distribuição:** Ubuntu 16.04 LTS

**Sublime Text 3:** Build 3114

**gcc version:** 5.3.1

**SQLite version:** 3.11.0

**PhpStorm:** 2016.1.2

**JRE:** 1.8.0\_76-release-b198 amd64

**PHP Version:** 7.0.4-7

**Composer version:** 1.1.2

**Laravel Framework version:** 5.2.39



### 3 Conclusão

O projeto desenvolvido apresenta uma estrutura sólida e modular. Apesar de alguns módulos terem uma contribuição mais significativa do que outros, todos eles são indispensáveis para o *setup* da plataforma final.

O resultado final corresponde às expectativas do grupo, e encontra-se pronto a ser utilizado por centros de investigação científica, como o IPATIMUP, e pelo meio académico como uma ferramenta de auxílio à investigação.

## 4 Melhoramentos

Uma funcionalidade útil que ainda não foi implementada é a possibilidade de ver os dados relativos a um único tecido reunidos num histograma com os valores dos coeficientes das correlações entre genes.

## 5 Recursos

### 5.1 Software

Ubuntu, <http://www.ubuntu.com/>

Sublime Text, <https://www.sublimetext.com/>

PhpStorm, JetBrains, <https://www.jetbrains.com/phpstorm/>

Composer, <https://getcomposer.org/>

Laravel, <https://laravel.com/>

jQuery, <https://jquery.com/>

SQLite, <https://www.sqlite.org/>

Bootstrap, <http://getbootstrap.com/>

clipboard.js, <https://clipboardjs.com/>

bootstrap-slider.js, <http://seiyria.com/bootstrap-slider/>

### 5.2 Outros

GTEEx Portal, <http://www.gtexportal.org/>

Stack Overflow, <http://stackoverflow.com/>

## A Apêndice

### A.1 Manual do utilizador

#### A.1.1 Execução da plataforma

*As instruções apresentadas de seguida assumem que o ambiente de desenvolvimento é uma distribuição Linux.*

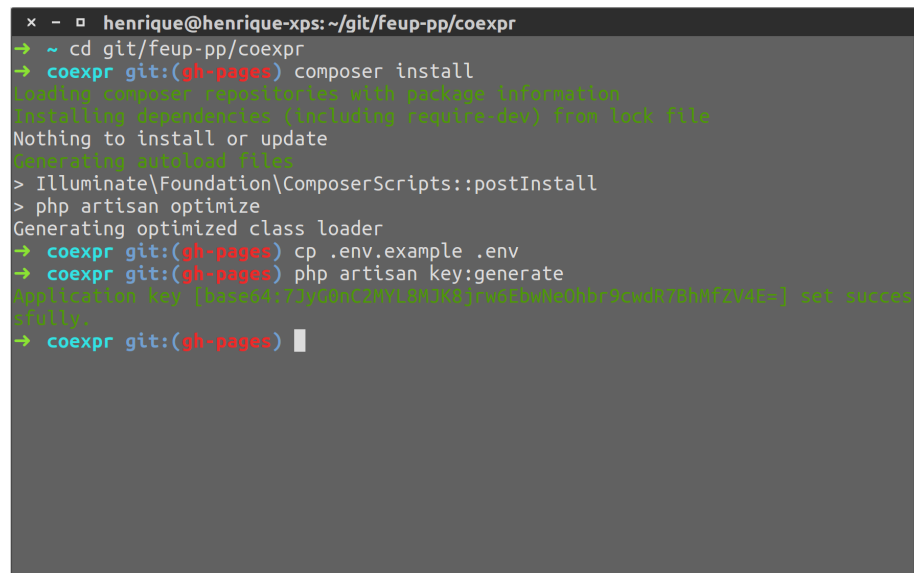
A plataforma foi feita com a framework de PHP [Laravel](#), portanto a primeira coisa a fazer é instalar o [Composer](#).

O próximo passo é descarregar o projeto, navegar até à pasta `coexpr`, e instalar as dependências com o comando:

```
composer install
```

De seguida, é preciso criar o ficheiro de ambiente `.env`, e gerar uma chave para a aplicação. Para isso, é necessário correr os seguintes comandos:

```
cp .env.example .env  
php artisan key:generate
```



```
x - □ henrique@henrique-xps: ~/git/feup-pp/coexpr  
→ ~ cd git/feup-pp/coexpr  
→ coexpr git:(gh-pages) composer install  
loading composer repositories with package information  
installing dependencies (including require-dev) from lock file  
Nothing to install or update  
generating autoload files  
> Illuminate\Foundation\ComposerScripts::postInstall  
> php artisan optimize  
Generating optimized class loader  
→ coexpr git:(gh-pages) cp .env.example .env  
→ coexpr git:(gh-pages) php artisan key:generate  
Application key [base64:7JyG8nE2MYL8PJk8jrw6EbwNeDfbr9cWdR7BhMf2V4E=] set successfully.  
→ coexpr git:(gh-pages) █
```

Figura 2: Screenshot da instalação do projeto através do terminal.

Finalmente, para lançar a plataforma basta correr o seguinte comando:

```
php artisan serve
```

Depois do comando anterior, ao navegar até <http://localhost:8000/>, o utilizador deverá conseguir ver a página principal da plataforma - tal como na figura 8.

Para terminar a execução local da plataforma, basta voltar ao terminal e interromper o comando pressionando as teclas `ctrl` + `C`.

### A.1.2 Preparação da base de dados

*Ao explorar um tecido na plataforma, a tabela desse tecido encontra-se vazia. Isso deve-se ao facto de a plataforma já estar operacional, mas a base de dados se encontrar vazia.*

*Esta secção serve para preparar a base de dados, para que possa então ser consultada através da plataforma.*

A primeira coisa a fazer é descarregar o [ficheiro comprimido](#) que contém toda a informação genética de que a plataforma precisa.

Assumindo, deste ponto em diante, que a estrutura de ficheiros do projeto é igual à da figura 3, o próximo passo é descompactar o ficheiro descarregado na pasta `project` ▶ `data` ▶ `input` ▶ `rna-seq-data`.

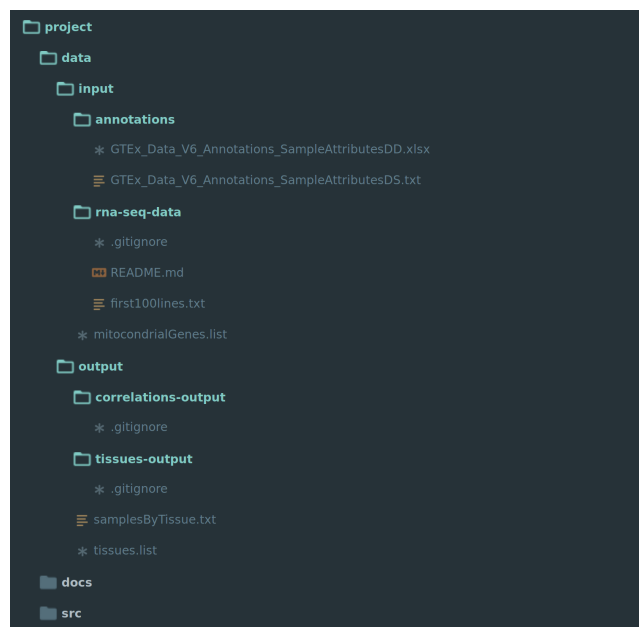
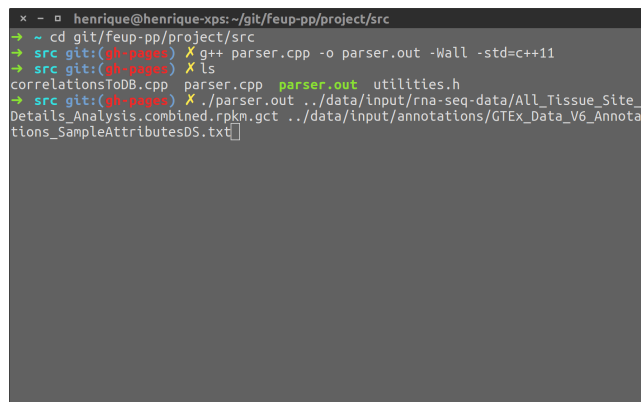


Figura 3: Estrutura de ficheiros do projeto que usa `C++` para preparar e preencher a base de dados que é utilizada pela plataforma.

Depois de o ficheiro descompactado ser colocado na pasta `rna-seq-data`, o passo seguinte é, através do terminal, navegar até à pasta `project` ▶ `src` e correr

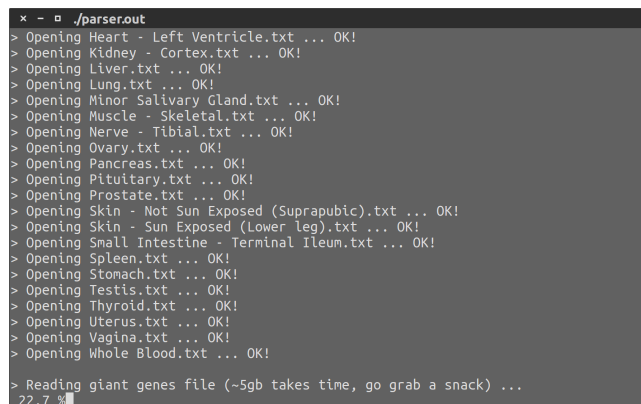
os seguintes comandos:

```
g++ parser.cpp -o parser.out -Wall -std=c++11
./parser.out ../data/input/rna-seq-data/All_Tissue_Site_Details_Analysis.combined.rpkm.gct ../data/input/annotations/GTEX_Data_V6_Annotations_SampleAttributesDS.txt
```



```
x - [henrique@henrique-xps:~/git/feup-pp/project/src]
→ ~ cd git/feup-pp/project/src
→ src git:(gh-pages) ✖ g++ parser.cpp -o parser.out -Wall -std=c++11
→ src git:(gh-pages) ✖ ls
correlationsToDB.cpp  parser.cpp  parser.out  utilities.h
→ src git:(gh-pages) ✖ ./parser.out ../data/input/rna-seq-data/All_Tissue_Site_Details_Analysis.combined.rpkm.gct ../data/input/annotations/GTEX_Data_V6_Annotations_SampleAttributesDS.txt
```

Figura 4: Execução dos dois comandos acima apresentados.



```
x - [./parser.out]
> Opening Heart - Left Ventricle.txt ... OK!
> Opening Kidney - Cortex.txt ... OK!
> Opening Liver.txt ... OK!
> Opening Lung.txt ... OK!
> Opening Minor Salivary Gland.txt ... OK!
> Opening Muscle - Skeletal.txt ... OK!
> Opening Nerve - Tibial.txt ... OK!
> Opening Ovary.txt ... OK!
> Opening Pancreas.txt ... OK!
> Opening Pituitary.txt ... OK!
> Opening Prostate.txt ... OK!
> Opening Skin - Not Sun Exposed (Suprapubic).txt ... OK!
> Opening Skin - Sun Exposed (Lower leg).txt ... OK!
> Opening Small Intestine - Terminal Ileum.txt ... OK!
> Opening Spleen.txt ... OK!
> Opening Stomach.txt ... OK!
> Opening Testis.txt ... OK!
> Opening Thyroid.txt ... OK!
> Opening Uterus.txt ... OK!
> Opening Vagina.txt ... OK!
> Opening Whole Blood.txt ... OK!

> Reading giant genes file (~5gb takes time, go grab a snack) ...
22.7 %
```

Figura 5: Output da execução do *parser* sobre o ficheiro descompactado. Este programa demora alguns minutos até terminar.

Após correr o *parser*, é necessário fazer *touch* às bases de dados. Para isso, basta correr o script *refreshDBs.sh*, e executar um comando adicional para criar as tabelas necessárias em cada base de dados:

```
sh refreshDBs.sh
php artisan migrate:refresh
```

```
x - henrique@henrique-xps:~/git/feup-pp/coexpr
→ ~ cd git/feup-pp/coexpr
→ coexpr git:(gh-pages) X sh refreshDBs.sh
→ coexpr git:(gh-pages) X php artisan migrate:refresh
Migration table not found.
Migration table created successfully.
Migrated: 2016_05_11_160118 create_correlations_tables
→ coexpr git:(gh-pages) X
```

Figura 6: Instância de um terminal onde se fez *refresh* às bases de dados. Após estes comandos, as bases de dados encontram-se inicializadas, mas vazias.

Depois disto, as bases de dados já estão prontas para serem preenchidas. Finalmente, para as preencher, é preciso compilar e correr o programa *correlationsToDB* para cada tecido.

```
g++ correlationsToDB.cpp -o correlationsToDB.out -Wall -lsqlite3 -std=c++11
./correlationsToDB.out "Bladder" ../data/input/mitochondrialGenes.list
```

```
x - henrique@henrique-xps:~/git/feup-pp/project/src
→ ~ cd git/feup-pp/project/src
→ src git:(gh-pages) X g++ correlationsToDB.cpp -o correlationsToDB.out -Wall -lsqlite3 -std=c++11
→ src git:(gh-pages) X ls
correlationsToDB.cpp correlationsToDB.out parser.cpp parser.out utilities.h
→ src git:(gh-pages) X ./correlationsToDB.out "Bladder" ../data/input/mitochondrialGenes.list
Reading Bladder file ... OK!
Opening DB ... OK!
Resetting DB table ... OK!
bladder ... OK!
Creating DB index (hang on just a bit more!) ... OK!
- All done! -
→ src git:(gh-pages) X
```

Figura 7: Resultado da compilação do programa *correlationsToDB*, e da execução do mesmo para o tecido *Bladder*.

Neste exemplo, após a execução dos comandos acima, o tecido *Bladder* está pronto para ser consultado através da plataforma. Se o utilizador navegar até

à página do tecido, a tabela já deverá ter a informação disponível, tal como é exemplificado na figura [10](#).



## A.2 Screenshots da plataforma

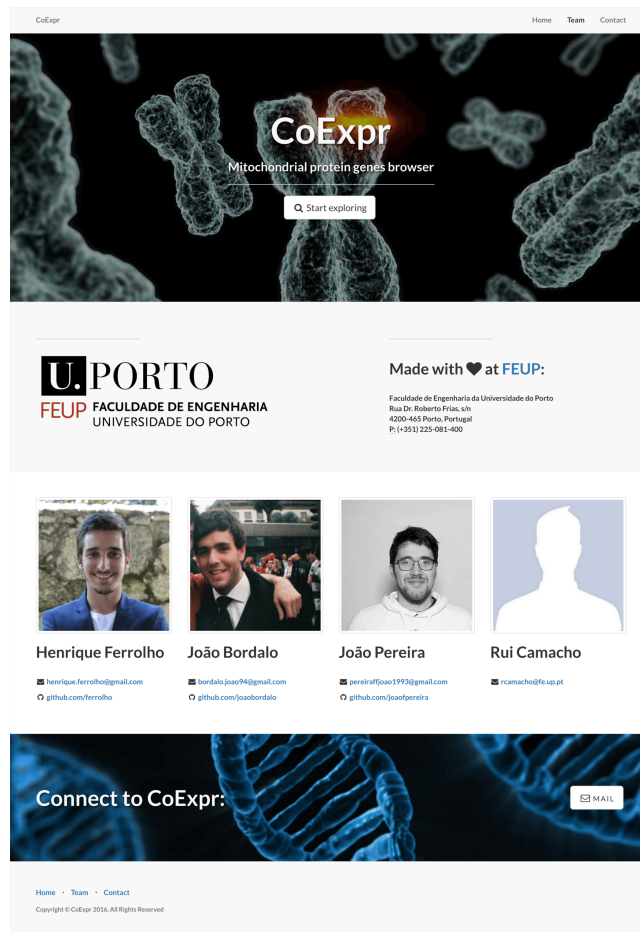


Figura 8: Página principal da plataforma online. O botão *Start exploring*, no topo, encaminha o utilizador para a página de seleção do tecido a explorar - figura 9.

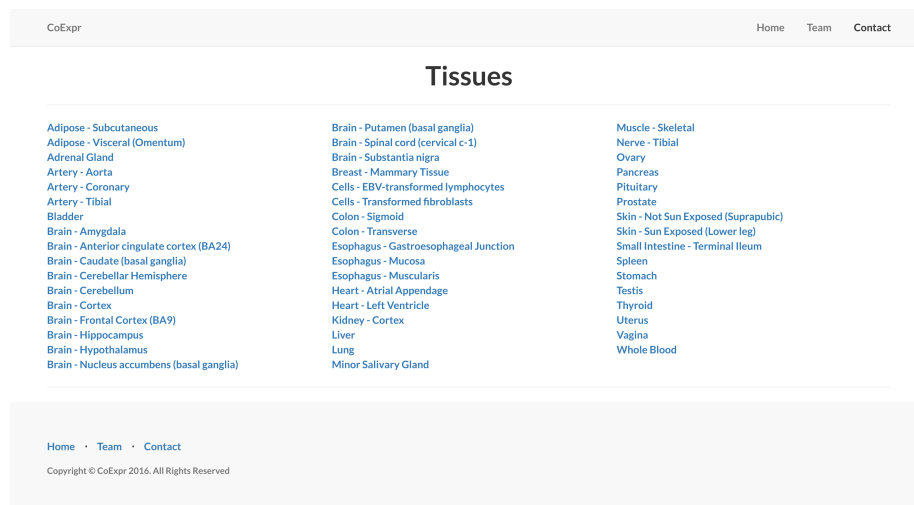


Figura 9: Página com a lista de tecidos disponíveis a serem explorados. Quando o utilizador carrega num dos tecidos, é encaminhado para a página específica de pesquisa nesse tecido - figura 10.

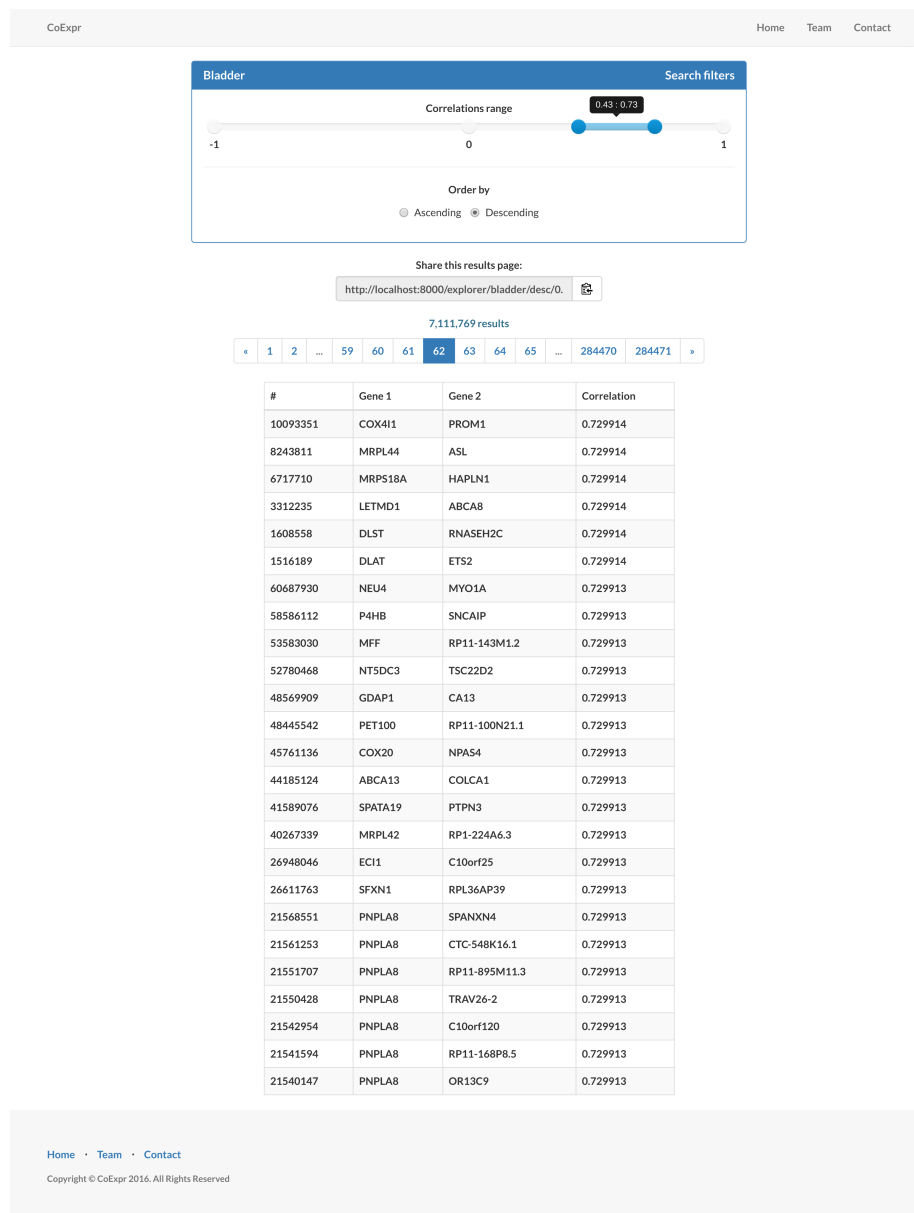


Figura 10: Página de exploração de um tecido (neste caso da bexiga). O painel, no topo, permite refinar os resultados da pesquisa. Diretamente abaixo do painel encontra-se uma widget para copiar para o clipboard o link dos resultados da pesquisa atual. Finalmente, a tabela mostra os resultados da pesquisa: o ID do par de genes na base de dados, cada um dos genes do par, e a semelhança desses genes.