

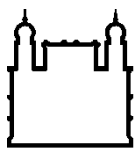
MINISTÉRIO DA SAÚDE
FUNDAÇÃO OSWALDO CRUZ
INSTITUTO OSWALDO CRUZ

Mestrado em Biologia Computacional e Sistemas

ANENDB: PREDIÇÃO COMPUTACIONAL E BANCO DE DADOS PARA
ENZIMAS ANÁLOGAS

ALEXANDER DA FRANCA FERNANDES

Rio de Janeiro
Dezembro de 2016



Ministério da Saúde

FIOCRUZ
Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

Programa de Pós-Graduação em Biologia Computacional e Sistemas

Alexander da Franca Fernandes

AnEnDB: predição computacional e banco de dados para enzimas análogas

Dissertação apresentada ao Instituto Oswaldo Cruz como parte dos requisitos para obtenção do título de Mestre em Biologia Computacional e Sistemas

Orientadores: Dra. Ana Carolina Ramos Guimarães
Dr. Marcos Paulo Catanho de Souza

RIO DE JANEIRO

Dezembro de 2016

Ficha catalográfica elaborada pela
Biblioteca de Ciências Biomédicas/ ICICT / FIOCRUZ - RJ

F363 Fernandes, Alexander da Franca

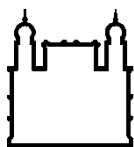
AnEnDB: predição computacional e banco de dados para enzimas
análogas / Alexander da Franca Fernandes. – Rio de Janeiro, 2016.
xvii, 104 f. : il. ; 30 cm.

Dissertação (Mestrado) – Instituto Oswaldo Cruz, Pós-Graduação em
Biologia Computacional e Sistemas, 2016.

Bibliografia: f. 91-104

1. Analogia enzimática. 2. Evolução. 3. Banco de dados. 4. KEGG.
5. Software. 6. AnEnDB. 7. AnEnPi. 8. Analogia. 9. Intergenômica. 10.
Intragenômica. I. Título.

CDD 572.7



Ministério da Saúde

FIOCRUZ

Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

Programa de Pós-Graduação em Biologia Computacional e Sistemas

AUTOR: ALEXANDER DA FRANCA FERNANDES

AnEnDB: predição computacional e banco de dados para enzimas análogas

**ORIENTADORES: Dra. Ana Carolina Ramos Guimarães
Dr. Marcos Paulo Catanho de Souza**

Aprovada em: ____/____/____

EXAMINADORES:

Dr. Antonio Basílio de Miranda - Presidente

(Laboratório de Biologia Computacional e Sistemas - IOC/FIOCRUZ)

Dr. Diogo Tschoek

(Laboratório de Microbiologia - Instituto de Biologia/UFRJ)

Dr. Sérgio Lifschitz

(Laboratório de Bioinformática PUC/RJ)

Dra. Adriana M. Fróes

(Laboratório de Microbiologia, Instituto de Biologia, Depto. de Biologia Marinha)

Dr. Fábio Mota

(Laboratório de Biologia Computacional e Sistemas - IOC/FIOCRUZ)

Rio de Janeiro, 12 de dezembro de 2016

À lucidez da ingenuidade.

AGRADECIMENTOS

Sem ele eu não saberia que a Bioinformática existia por perto:

Ao Wim Degrave. Esteve presente em muitas conversas antes mesmo de eu sequer supor entrar na vida acadêmica.

Sem eles eu não teria sequer o que fazer ou onde fazer:

Toda a equipe da Pós-Graduação da Fiocruz.

Sem ele eu não conheceria a porta de entrada para a vida acadêmica:

Alex Amorim, pelo gesto de amizade inesquecível. Foi quem ouviu minhas queixas e me levou até meus orientadores.

Sem ela eu não teria entrado pela porta da vida acadêmica:

Cris Lobo me fez alcançar o que parecia impraticável. Foi uma revolução nosso primeiro encontro, foi uma revolução o último.

Sem eles eu não teria cursado um único dia sequer:

Laura Barreira, Daniela Galper, Judy Galper e todos da Escola EDEM, pelo apoio que beira o inexplicável de tão raro na história de qualquer indivíduo.

Sem eles eu não teria conseguido seguir o caminho até aqui:

Carol e Catanho, pelo acolhimento enorme, com carinho, vigor, conhecimento, juventude, inspiração e alegria. Por terem sido os gênios mágicos que me aconselharam durante todo o processo. Transformaram minha vida para melhor em diversidade e profundidade.

Sem ela eu não teria entusiasmo para vislumbrar o depois:

Laís, por ser a gasolina em quantidades de *rock star* que apareceu no final do caminho, justo quando o combustível já havia acabado e tudo parecia destinado a um fim monocórdico, enfadonho e sem perspectivas. Se é possível *hackear* um plano de vida que já estava solidamente definido... você conseguiu.

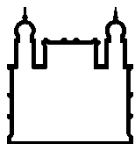
Sem eles eu não teria memórias para levar comigo:

Aos meus amigos da 201: Fabio Passetti, Márcio, Vanessa, Rafael, Phillippe, Tavares e Gabriel. É um grupo inesquecível, aprendi muito com vocês em diversos aspectos, acadêmicos e da vida. Vocês foram de fato inspiradores num nível que talvez não conheçam. Contem comigo sempre.

Agradeço a todos como um reconhecimento de que não existe qualquer êxito ou futuro na vida sem a força de outras pessoas.

*“For us, there is no spring.
Just the wind that smells fresh before the
storm”.*

Conan, o cimério



Ministério da Saúde

FIOCRUZ

Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

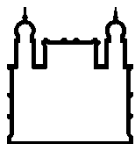
AnEnDB: predição computacional e banco de dados para enzimas análogas

RESUMO

DISSERTAÇÃO DE MESTRADO EM BIOLOGIA COMPUTACIONAL E SISTEMAS

Alexander da Franca Fernandes

O AnEnDB é uma ferramenta e um sistema de banco de dados especializado em enzimas análogas. As enzimas análogas originam-se a partir de eventos evolutivos independentes, convergindo para uma mesma (ou similar) função biológica e/ou possuem distintos mecanismos de catálise. Investigações sobre a ocorrência de enzimas análogas em vias metabólicas podem não somente ampliar a compreensão sobre a origem e evolução das vias bioquímicas como também revelar novos alvos para o desenvolvimento de fármacos. Muitas vezes tais eventos são ignorados e/ou subestimados devido aos próprios critérios de busca e seleção destes alvos, usualmente baseados na especificidade de funções enzimáticas e não na origem evolutiva das diferentes formas de uma determinada enzima. Alguns trabalhos sugerem que a fração de atividades enzimáticas nas quais ocorreram múltiplos eventos de origem independente pode ser substancial. Contudo, este é um tema ainda pouco explorado e, até o momento, um estudo global da ocorrência, distribuição e implicações destes eventos, envolvendo os organismos cujos genomas foram completamente sequenciados, ainda não foi realizado. O AnEnDB é capaz de auxiliar análises de enzimas análogas em diferentes organismos, através de uma ferramenta web de acesso público contendo uma nova versão do *pipeline* para predição computacional de enzimas análogas (AnEnPi-v2) e um sistema de banco de dados de sequência, estrutura e evolução de enzimas análogas. Este sistema deverá ser capaz de responder diferentes questões biológicas relacionadas à analogia funcional.



Ministério da Saúde

FIOCRUZ

Fundação Oswaldo Cruz

INSTITUTO OSWALDO CRUZ

AnEnDB: computational prediction and database for analogous enzymes

ABSTRACT

TESIS IN COMPUTATIONAL BIOLOGY AND SYSTEMS

Alexander da Franca Fernandes

AnEnDB is a software and a relational database designed for the analysis of analogous enzymes. Analogous enzymes arise from independent evolutionary events, converging for a similar biological function, and may possess different catalytic mechanisms as well. Investigations on the occurrence of analogous enzymes in metabolic pathways can not only broaden our understanding of the origin and evolution of biochemical pathways but also reveal new targets for drug development, often overlooked and/or underestimated due to the criteria for searching and selecting these targets, usually based on specificity and enzymatic functions and not on the evolutionary origin of distinct forms of a given enzyme. Several studies suggest that the fraction of enzymatic activities in which multiple events of independent origin have occurred during evolution is substantial. However, this subject is still poorly understood, and a comprehensive investigation of the occurrence, distribution and implications of these events, involving organisms whose genomes have been completely sequenced, has not been accomplished so far. AnEnDB assists the analysis of analogous enzymes in different organisms, providing a publicly accessible web tool based on a new version of the pipeline for computational prediction of analogous enzymes (AnEnPi-v2) and a sequence, structure and evolution database of analogous enzymes. AnEnDB system should be able to answer different biological questions related to functional analogy.

ÍNDICE

INTRODUÇÃO	1
Bioinformática	1
Enzimas e suas atividades funcionais	1
Metabolismo	3
Homologia	5
Analogia	6
Bancos de Dados Biológicos	8
Permanência de bancos de dados	9
Kyoto Encyclopedia of Genes and Genomes (KEGG)	10
Structural Classification Of Proteins database (SCOP)	10
Protein Data Bank (PDB).....	11
Bancos de Dados Relacionais.....	12
Metodologia de desenvolvimento de <i>software</i>	13
OBJETIVOS	15
Objetivo Geral	15
Objetivos Específicos	15
MATERIAL E MÉTODOS	16
Metodologia de desenvolvimento de <i>software</i>	16
Divisão do projeto em áreas de estudo e atuação	17
Origem de dados	18
Algoritmo de agrupamento de enzimas análogas	20
Banco de Dados Relacional	22
Ambiente de Desenvolvimento	22
Módulo para abstração de banco de dados	25

Ferramenta para documentação de código.....	27
Framework para desenvolvimento web.....	29
Apresentação de dados	30
Interface web.....	30
REpresentational State Transfer (REST)	32
RESULTADOS E DISCUSSÃO	33
Acesso ao AnEnDB	33
<i>Metodologia de Desenvolvimento de Software</i>.....	34
<i>Banco de dados relacional</i>.....	36
AnEnDB: codificação	40
Modelagem e generalização da origem primária de dados.....	41
Processamento de dados e informações no contexto de enzimas	
análogas.....	43
Apresentação dos dados através de dois meios distintos.	44
Interface web.....	44
REST: URLs diretas que retornam resultados de pesquisa	45
AnEnDB: exploração dos dados	46
Estudo de caso: analogia intergenômica entre <i>Trypanosoma cruzi</i> e	
<i>Homo sapiens</i>.....	57
Estudo de caso: analogia intragenômica em <i>Homo sapiens</i>.....	65
Validação de Dados	71
CONCLUSÃO	73
REFERÊNCIAS BIBLIOGRÁFICAS.....	75

LISTA DE FIGURAS

FIGURA 1. MAPA REPRESENTANDO A VIA GLICOLÍTICA EM HUMANOS.....	4
FIGURA 2. REPRESENTAÇÃO DA ESTRUTURA TRIDIMENSIONAL DA ENZIMA PIRUVATO CINASE M2 OBTIDA NO <i>PROTEIN DATA BANK</i> (PDBid: 3BJT).	12
FIGURA 3. DEMONSTRAÇÃO GRÁFICA DE ORGANIZAÇÃO DE TABELAS EM UM BANCO DE DADOS RELACIONAL.....	13
FIGURA 4. <i>PRODUCT BACKLOG</i> . FERRAMENTA UTILIZADA EM <i>SCRUM</i> PARA ACOMPANHAMENTO DE TAREFAS.	16
FIGURA 5. DESENHO EXPERIMENTAL DO ANENDB.	17
FIGURA 6. PARTE DO CONTEÚDO DO ARQUIVO MID_ENZYME.LIST DO KEGG.	18
<i>FIGURA 7. PARTE DO CONTEÚDO DO ARQUIVO T02325.PEP, DO KEGG</i>	19
FIGURA 8: DIAGRAMA DEMONSTRANDO A RELAÇÃO ENTRE CAMADA DE <i>EXTRATORES DE DADOS</i> , ORIGEM DE DADOS E APRESENTAÇÃO DOS DADOS.....	20
FIGURA 9. DIAGRAMA DEMONSTRANDO A METODOLOGIA DE AGRUPAMENTO DO ANENDB.	21
FIGURA 10. COMPARAÇÃO ENTRE AS BUSCAS POR “PYTHON” (CURVA AZUL) E “PERL” (CURVA VERMELHA) NO MUNDO A PARTIR DO GOOGLE TRENDS.	24
FIGURA 11: COMPARAÇÃO ENTRE AS BUSCAS POR “PYTHON” (CURVA AZUL) E “PERL” (CURVA VERMELHA) NO BRASIL A PARTIR DO GOOGLE TRENDS.	24
FIGURA 12. EXEMPLO DE SCRIPT PYTHON PARA EFETUAR UMA OPERAÇÃO EM UM BANCO DE DADOS RELACIONAL (SEM A UTILIZAÇÃO DE ORM).....	26
FIGURA 13. EXEMPLO DE SCRIPT PYTHON PARA EFETUAR UMA OPERAÇÃO EM UM BANCO DE DADOS RELACIONAL (COM A UTILIZAÇÃO DE ORM).....	26
<i>FIGURA 14. EXEMPLO DE CÓDIGO DE PÁGINA HTML QUE LISTA TODOS OS ORGANISMOS DO ANENDB.</i>	30
FIGURA 15. REPRESENTAÇÃO DO FLUXO DE UMA PÁGINA WEB DINÂMICA.....	31
FIGURA 16. REPRESENTAÇÃO DO FLUXO DE UMA PÁGINA WEB DINÂMICA UTILIZANDO O FRAMEWORK FLASK E WSGI.	31
FIGURA 17. PRIMEIRA PÁGINA (HOME) DA INTERFACE WEB DO ANENDB.	33
FIGURA 18. DIAGRAMA QUE REPRESENTA OS RELACIONAMENTOS ENTRE AS PRINCIPAIS TABELAS DO ANENDB.	38
FIGURA 19. DIAGRAMA QUE REPRESENTA AS TABELAS QUE REGISTRAM OS GRUPOS DE ANÁLOGOS (CLUSTERS) E OS VALORES OBTIDOS NA BUSCA POR SIMILARIDADE DE SEQUÊNCIA (SIMILARITIES).	39
FIGURA 20. CAMADAS DO ANENDB.	40

FIGURA 21. EXEMPLO DE DADO PARA A SEQUÊNCIA DE AMINOÁCIDOS DA PROTEÍNA MMA:MM_2626 DO ARQUIVO T00082.PEP DO KEGG.	41
FIGURA 22. PARTE DO CONTEÚDO DO ARQUIVO, DO KEGG, MMA_ENZYME.LIST.	42
FIGURA 23. ESTRUTURA DE DADOS <i>PROTEIN</i> , GERADA E UTILIZADA PELO ANENDB.	42
FIGURA 24. EXEMPLO DE UTILIZAÇÃO DA INTERFACE REST.	46
FIGURA 25. EXEMPLO DE CÓDIGO QUE INSTANCIA, NA VARIÁVEL ANENDB, A CLASSE ANENDB.	47
FIGURA 26. EXEMPLO DE CÓDIGO, UTILIZANDO O OBJETO ANENDB,	47
FIGURA 27. EXEMPLO DE UTILIZAÇÃO DO CÓDIGO DO ANENDB	48
FIGURA 28. EXEMPLO DE UTILIZAÇÃO DO CÓDIGO DO ANENDB PARA OBTER O TOTAL DE ORGANISMOS REGISTRADOS NO BANCO DE DADOS RELACIONAL.	48
FIGURA 29: DIFERENÇA NO VOLUME DE DADOS DO KEGG ENTRE OS ANOS DE 2006 E 2016.	49
FIGURA 30. EXEMPLO DA UTILIZAÇÃO DO MÉTODO GETORGANISMBYNAME QUE RETORNA INFORMAÇÕES SOBRE A TAXONOMIA DE ORGANISMOS.	50
FIGURA 31. EXEMPLO DE UTILIZAÇÃO DO MÉTODO GETTOTALGROUPSOFHOMOLOGOUS.	50
FIGURA 32. EXEMPLO DE CÓDIGO DO ANENDB QUE RETORNA AS CLASSES DE ATIVIDADES ENZIMÁTICAS QUE POSSUEM MAIS GRUPOS (CLUSTERS).	50
FIGURA 33. DISTRIBUIÇÃO ENTRE NÚMEROS EC COM TODOS OS NÍVEIS DE CLASSIFICAÇÃO ENZIMÁTICA DEFINIDOS (COMPLETOS) OU INCOMPLETOS.	52
FIGURA 34. EXEMPLO DE CÓDIGO DO ANENDB QUE PODE SER UTILIZADO PARA INFORMAR O TOTAL DE ENZIMAS DE UMA CLASSE DE ATIVIDADE ENZIMÁTICA.	52
FIGURA 35. EXEMPLO, RESUMIDO, DA INFORMAÇÃO RETORNADA PELO MÉTODO GETORGANISMINTRAGENOMICANALOGY	53
FIGURA 36. EXEMPLO DE CÓDIGO DO ANENDB QUE RETORNA O TOTAL DE ORGANISMOS QUE POSSUEM ANALOGIA INTRAGENÔMICA.	53
FIGURA 37. RELAÇÃO, POR DOMÍNIO, ENTRE ORGANISMOS QUE POSSUEM ANALOGIA INTRAGENÔMICA EM RELAÇÃO AO TOTAL DE ORGANISMOS REGISTRADOS NO KEGG.	54
FIGURA 38. RELAÇÃO DE ANALOGIA INTRAGENÔMICA EM COMPARAÇÃO AO TOTAL DE ORGANISMOS REGISTRADOS NO KEGG, AGRUPADOS POR REINOS DO DOMÍNIO EUCARIOTO.	55
FIGURA 39. PERCENTUAL DE ANALOGIA INTRAGENÔMICA, A PARTIR DOS ORGANISMOS REGISTRADOS NO KEGG, NOS DOMÍNIOS EUCARIOTO, EUBACTÉRIA E ARCHAEA.	55

FIGURA 40. EXEMPLO DE CÓDIGO QUE BUSCA EM TODAS AS CLASSES DE ATIVIDADES ENZIMÁTICAS AQUELAS QUE POSSUEM APENAS UM GRUPO (CLUSTER).	56
FIGURA 41. DISTRIBUIÇÃO ENTRE ATIVIDADES ENZIMÁTICAS (REPRESENTADAS POR NÚMEROS EC) NAS QUAIS APENAS UM ÚNICO GRUPO DE ENZIMAS FOI FORMADO APÓS O AGRUPAMENTO PELO PIPELINE ANENPI-V2.....	56
FIGURA 42. EXEMPLO DE UM RESULTADO DE DA BUSCA FEITA ATRAVÉS DA INTERFACE WEB DO ANENDB	57
FIGURA 43. EXEMPLO DE CÓDIGO DO ANENDB QUE RETORNA DADOS DOS ORGANISMOS QUE POSSUEM NO NOME CIENTÍFICO A STRING “CRUZI”	58
FIGURA 44. EXEMPLO DE CÓDIGO DO ANENDB QUE RETORNA OS DADOS DOS ORGANISMOS QUE POSSUEM NO NOME CIENTÍFICO A STRING “HOMO”	59
FIGURA 45. TELA QUE MOSTRA O RESULTADO DE DA BUSCA PELA ATRAVÉS DA INTERFACE WEB DO ANENDB PARA ORGANISMOS CUJO NOME CIENTÍFICO QUE POSSUEM POSSUA A STRING HOMO.	59
FIGURA 46. TELA DA INTERFACE WEB DO ANENDB MOSTRANDO A OPÇÃO DE EXECUTAR UMA BUSCA POR ANALOGIA INTERGENÔMICA.	60
FIGURA 47. EXEMPLO DE CÓDIGO DO ANENDB QUE RETORNA AS CLASSES DE ATIVIDADE ENZIMÁTICA COM ANALOGIA INTERGENÔMICA	60
FIGURA 48. LISTA DE CLASSES DE ATIVIDADES ENZIMÁTICAS COM ANALOGIA (EXCLUSIVAMENTE) INTERGENÔMICA ENTRE <i>H. SAPIENS</i> E <i>T. CRUZI</i>	61
FIGURA 49. LISTA DE CLASSES DE ATIVIDADES ENZIMÁTICAS COM ANALOGIA (EXCLUSIVAMENTE) INTERGENÔMICA ENTRE <i>H. SAPIENS</i> E <i>T. CRUZI</i> ,	61
FIGURA 50. VIA METABÓLICA GLICÓLISE/GLICONEOGÊNESE REPRESENTANDO AS ATIVIDADES ENZIMÁTICAS ANOTADAS NOS GENOMAS DOS ORGANISMOS <i>H. SAPIENS</i> E <i>T. CRUZI</i>	63
FIGURA 51. GRUPOS (CLUSTERS) AOS QUAIS PERTENCEM AS FORMAS ANÁLOGAS ENTRE <i>H. SAPIENS</i> E <i>T. CRUZI</i> NA ATIVIDADE ENZIMÁTICA 2.7.1.2 DA VIA GLICOLÍTICA.	64
FIGURA 52. ÚNICA SEQUÊNCIA DO CLUSTER NÚMERO 4142 DO ORGANISMO <i>T. CRUZI</i> DA CLASSE DE ATIVIDADE ENZIMÁTICA 2.7.1.2.	64
FIGURA 53. EXEMPLO DE CÓDIGO DO ANENDB QUE RETORNA AS CLASSES DE ATIVIDADE ENZIMÁTICA QUE POSSUEM ANALOGIA INTRAGENÔMICA.....	66
FIGURA 54. A BUSCA POR ORGANISMOS APRESENTA A LISTA DE EC QUE POSSUEM ANALOGIA INTRAGENÔMICA.	67
FIGURA 55. EXEMPLO DE EXPLORAÇÃO, ATRAVÉS DO CÓDIGO DO ANENDB, DOS DADOS DE ANALOGIA INTRAGENÔMICA PARA A CLASSE DE ATIVIDADE ENZIMÁTICA 5.3.99.2.	68

FIGURA 56. TELA DO ANENDB PARA INICIAR BUSCA POR ANALOGIA INTRAGENÔMICA.	68
FIGURA 57. RESULTADO MOSTRANDO A LISTA DE CLASSES DE ATIVIDADE ENZIMÁTICA QUE POSSUEM ANALOGIA INTRAGENÔMICA EM H. SAPIENS	69
FIGURA 58. TELA DO ANENDB QUE EXIBE OS GRUPOS DE ENZIMAS DAS CLASSES DE ATIVIDADE ENZIMÁTICA 2.7.7.7 E 3.1.3.2 PARA O ORGANISMO <i>H. SAPIENS</i>	70
FIGURA 59. TELA DO ANENDB EXIBINDO AS SEQUÊNCIAS DO GRUPO 10 DA CLASSE DE ATIVIDADE ENZIMÁTICA 2.7.7.7, EM <i>H.</i> <i>SAPIENS</i>	70

INTRODUÇÃO

Bioinformática

A Bioinformática é um campo de estudo interdisciplinar que envolve diversas áreas, tais como a Biologia, a Ciência da Computação, a Matemática e a Estatística (1) podendo ser definida como a pesquisa, o desenvolvimento ou a aplicação de técnicas ou ferramentas computacionais para adquirir, armazenar, organizar, analisar, visualizar e integrar dados e informações de origem biológica, médica, comportamental ou de saúde (2). A Bioinformática lida com os desafios da Biologia aplicando diversas técnicas como processamento de imagem, simulações computacionais, análise de redes, mineração de dados, entre outras, para realizar estudos de genômica comparativa, análise de expressão gênica, análise estrutural de proteínas, filogenética, redes metabólicas, citando apenas algumas dessas aplicações. Apesar dos desafios, as aplicações em Bioinformática são uma realidade e já são parte, por exempl, de processos importantes de saúde pública de nível mundial, como no auxílio ao controle de epidemias como o *influenza* (3) e do Zika vírus (4). Igualmente, os profissionais da área de bioinformática são requisitados em grupos de diversas áreas de pesquisa (5). Não apenas na área médica, a Bioinformática tem também contribuições importantes na agricultura (6) para o desenvolvimento de plantas mais resistentes (7), indústria, para a produção de substâncias minerais a partir de microrganismos (8) e ecologia, a partir de pesquisas relacionadas ao uso sustentável de recursos naturais (9). Portanto, a Bioinformática tem um papel central nas novas descobertas científicas no mundo atual, atendendo a uma grande demanda por novas técnicas, *softwares* e modelos de análise computacional.

Enzimas e suas atividades funcionais

Enzimas são proteínas que possuem funções catalizadoras nos organismos, ou seja, aceleram reações químicas. Suas atividades permitem aceleração de reações na ordem de milhões de vezes ou mais. A enzima anidrase carbônica, por exemplo, é capaz de acelerar a transferência de CO₂ dos tecidos para a circulação

sanguínea numa velocidade 10^7 vezes maior do que seria possível sem a sua presença (10). Por seu papel central no metabolismo as enzimas são fundamentais para a existência da vida como se conhece. Sem o surgimento destes catalizadores durante a evolução, a lentidão das reações químicas que ocorrem naturalmente inviabilizaria o surgimento e evolução dos seres vivos em nosso planeta.

As enzimas são moléculas altamente especializadas tanto nas reações que realizam quanto na afinidade por substratos. Isso se deve à interação precisa (enzima e substrato) entre suas estruturas tridimensionais (10). Um conceito geral que descreve como as enzimas realizam suas reações químicas define que estas moléculas possuem um ou mais sítios catalíticos onde a reação específica ocorre. Essas reações foram divididas em seis categorias: 1) oxireductases, que catalisam reações de oxidação/redução (transferência de elétrons entre átomos de hidrogênio ou oxigênio); 2) transferases, assim chamadas por transferirem um grupo funcional entre duas moléculas; 3) hidrolases, que catalisam a quebra de ligações covalentes com a utilização de molécula de água; 4) liases: adicionam ou removem grupos químicos de substratos (sem hidrólise); 5) isomerases: catalisam um rearranjo intramolecular; e 6) ligases, que unem duas moléculas pela síntese de ligações C-O, C-S, C-N ou C-C.

Para vários tipos de reações catalisadoras as enzimas necessitam da participação de outras moléculas chamadas cofatores. Cofatores podem ser íons metálicos ou moléculas orgânicas, neste último caso os cofatores são chamadas de coenzimas. Cofatores participam da catálise das reações exercendo diferentes funções, como por exemplo: completar ou modificar o sítio ativo de uma enzima, doar elétrons ou átomos para o substrato, ou polarizar o sítio ativo da enzima onde a reação enzimática é executada. Não apenas cofatores e coenzimas regulam a atividade enzimática, mas outros fatores também atuam para viabilizar, influenciar a eficácia ou mesmo inibir totalmente a reação catalítica de uma enzima: temperatura, pH, concentração do substrato que reage com a enzima e moléculas inibidoras (10).

Inicialmente a nomenclatura enzimática não incluía nenhuma informação sobre os substratos que utilizam ou as reações que catalisam. Com o rápido crescimento na identificação de novas enzimas surgiu a necessidade de sistematizar tanto a nomenclatura das enzimas quanto suas reações catalisadoras. Dessa forma, em 1956 a *International Union of Biochemistry* estabeleceu a *International Commission on Enzymes* e já em 1958 as enzimas foram então divididas em 6

categorias principais, como visto anteriormente. Cada uma das seis categorias foram posteriormente subdivididas, sendo que cada enzima recebeu um código único de quatro dígitos, conhecido então como número *Enzyme Commission*, ou simplesmente número EC (11). Neste sistema de classificação o primeiro dígito de um número EC representa a classe química da enzima (conforme exposto anteriormente: oxido-redutase, transferase etc) enquanto o segundo e terceiro dígitos indicam geralmente o grupo químico envolvido na reação. Por fim, o quarto dígito informa a especificidade do substrato e/ou cofatores. Por exemplo, o EC 2.7.1.40 (piruvato cinase) representa as enzimas transferases (EC 2) que atuam na transferência de grupos que contém fósforo (EC 2.7) com álcool como acceptor (EC 2.7.1) e como substratos o piruvato e ATP (EC 2.7.1.40) (12).

Metabolismo

Metabolismo é essencialmente uma série de reações químicas interconectadas que se iniciam em uma molécula até a sua conversão em uma ou mais moléculas diferentes (10), podendo ser representado através de uma rede complexa de reações químicas chamadas vias metabólicas, como ilustrado na Figura 1. Essas vias possuem interconexões com outras vias, assim como produtos metabólicos destas vias podem servir como substrato para outras vias. É através dessa rede de reações químicas que as células obtêm energia e sintetizam suas macromoléculas (10). A ação das vias metabólicas pode ser percebida cotidianamente, como por exemplo, quando na execução de exercício físico intenso, o ácido láctico produzido concentra-se nas fibras musculares provocando fadiga, ao passo que o glicogênio, forma de carboidrato armazenado no tecido muscular, é consumido neste (13).

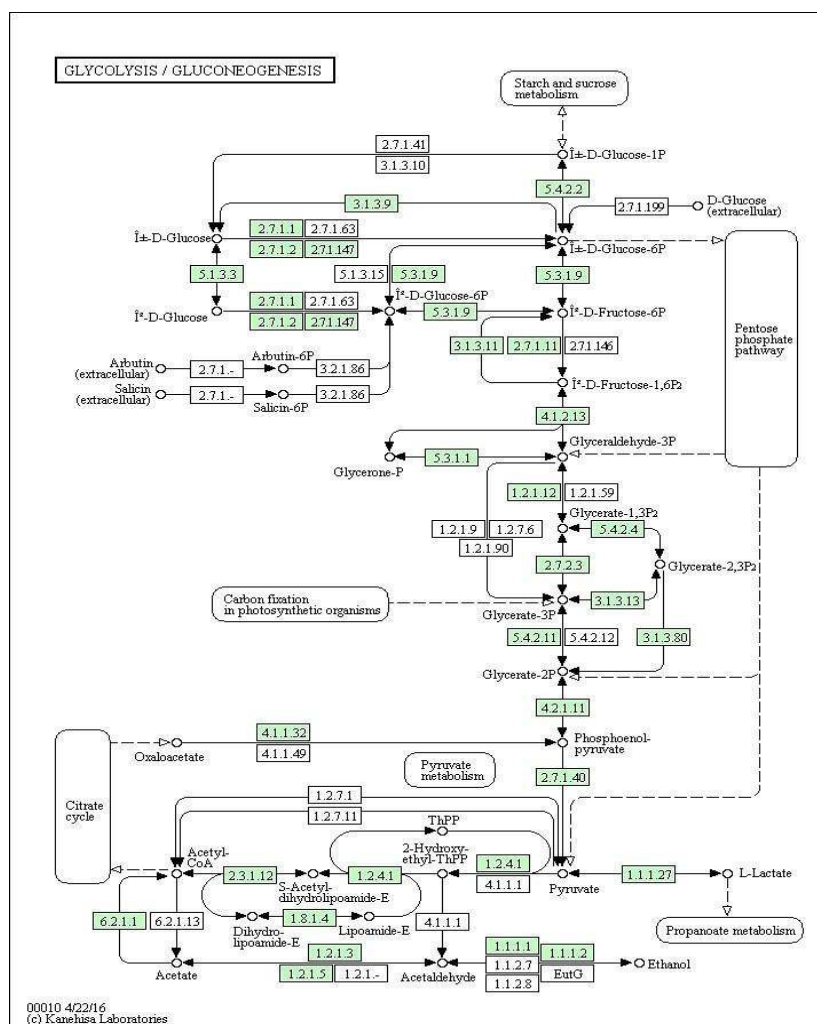


Figura 1. Mapa representando a via glicolítica em humanos. Os retângulos em cor verde representam as atividades enzimáticas presentes em *Homo sapiens*, os demais retângulos representam atividades enzimáticas existentes na via glicolítica, porém, ausentes no genoma humano. Fonte: <http://www.genome.jp/kegg/>.

Os elementos centrais nas vias metabólicas são as enzimas. São elas que catalisam as reações químicas entre substrato e produto e sem as quais diversas substâncias necessárias para a sobrevivência não seriam produzidas em quantidade suficiente, ou no tempo necessário, ou ainda, sequer seriam produzidas sem a presença de enzimas.

A compreensão sobre o funcionamento das diversas vias metabólicas é fundamental para muitas áreas como, por exemplo, a Biologia Sintética, que aplica

princípios de Engenharia em Biologia. A Biologia Sintética possui diversas aplicações, por exemplo, em produção de energia (biocombustíveis), medicina (produção de medicamentos), meio ambiente (agrotóxicos menos prejudiciais ou mais eficientes), bem como em diferentes abordagens dedicadas à construção de genomas mínimos (mínimo de genes necessários para um organismo) e protocélulas (células produzidas sinteticamente). Dentre as abordagens existentes em Biologia Sintética a engenharia metabólica se propõe a modificar o metabolismo de organismos e obter produtos que não seriam gerados naturalmente por estes. Apesar da complexidade em produzir resultados úteis com a modificação de vias metabólicas, existem estudos que resultaram na produção de substâncias de valor industrial como 1,4-butanediol (8) (polímero que normalmente é obtido apenas através de fontes minerais) e aminoácidos aromáticos (14), ambos utilizando *Escherichia coli* como organismo de expressão dessas substâncias. Para esses estudos, portanto, foi necessário não apenas compreender o funcionamento das vias metabólicas alvos, mas igualmente ser capaz de detalhar o metabolismo da bactéria *Escherichia coli*.

Além da Biologia Sintética, a Medicina utiliza o estudo de vias metabólicas a fim de encontrar terapias para doenças metabólicas como a adrenoleucodistrofia (distúrbio metabólico que provoca o acúmulo excessivo de ácidos graxos no cérebro) (15), compreender alguns mecanismos de patógenos importantes como *Mycobacterium tuberculosis* (16), ou até mesmo encontrar possíveis novas terapias para câncer ao partir do conhecimento do metabolismo de alguns tipos de células cancerígenas (17).

Homologia

Homologia é a relação entre dois caracteres que descendem, usualmente por divergência evolutiva, de um caracter ancestral comum (18). No que se refere às sequências biológicas, a homologia é comumente inferida com base no grau de similaridade medido entre pares de sequências. Sendo assim, apesar de ser um conceito claro, não é possível determinar a relação de ancestralidade entre sequências com total objetividade. Para tanto, seria necessário definir qual/quais parâmetro(s) de similaridade e suas medidas (valores) seriam capazes de distinguir

inequivocamente homólogos de não homólogos, bem como excluir a possibilidade de analogia (convergência a partir de caracteres ancestrais não relacionados) (18). Apesar de, na prática, não ser possível afirmar em termos absolutos a homologia entre sequências, este não é um conceito que quantifica a aproximação de ancestralidade, ou seja, sequências não podem ser apresentadas em termos de percentual de homologia; sequências são homólogas ou não, não havendo ponderação quantitativa sobre essa característica. Por outro lado, é possível apresentar a comparação entre sequências em termos de percentual de similaridade (maior ou menor), de acordo com o critério escolhido (18).

Entre genes, a homologia pode assumir duas categorias distintas: paralogia e ortologia. Parálogos são genes que derivam de duplicação gênica. A palavra “parálogo” advém do termo “paralelo” por indicar que o gene evoluiu em paralelo dentro da espécie (19). Duplicação gênica é um importante mecanismo de aquisição de novos genes e geração de novas funções nos organismos, provocada por diferentes fenômenos evolutivos como, por exemplo, *crossing over* desigual; retrotransposição, quando um gene já transcrito para RNA é reversamente transcrito e reinserto no DNA; e quando ocorre a duplicação completa do genoma (20). Por outro lado, ortólogos são genes em diferentes espécies que surgiram a partir de um único gene de uma espécie imediatamente ancestral a elas. A distinção de homólogos entre ortólogos e parálogos é crucial para descrever relações evolutivas com maior precisão. Também é importante para inferir a função de um gene (embora a conservação da função de um gene ortólogo não seja parte de sua própria definição, mas é normalmente uma consequência) (21).

Analogia

Outro conceito fundamental em evolução, além de homologia, é o de analogia. São dois os principais processos que levam ao surgimento de características análogas: convergência evolutiva e evolução paralela. Em evolução paralela características similares surgem a partir de linhagens diferentes e próximas de organismos como, por exemplo, a capacidade de planar entre alguns mamíferos (esquilos, lêmures e marsupiais). Já em convergência evolutiva, similaridades surgem de características distintas, em diferentes e distantes linhagens, como

resposta de adaptação a similares ambientes e similares estratégias de sobrevivência, como por exemplo, o surgimento de asas em aves e em insetos (19). Convergência evolutiva implica em linhagens distantes parecerem mais próximas do que realmente são (22), um exemplo claro é a semelhança entre baleias (mamíferos) e peixes. Apesar de a identificação de convergência em nível morfológico ser uma tarefa já com grandes desafios, no nível molecular a identificação de eventos de analogia implica em desafios mais intrincados e com regras que devem ser mais bem definidas (22). Os eventos de convergência em nível molecular podem ser classificados como convergência funcional (ocorre quando uma função molecular - atividade enzimática, por exemplo - surge de forma independente em mais de uma ocasião); convergência mecanística (quando proteínas possuem estruturas tridimensionais e sequências distintas, porém com o mesmo mecanismo catalítico); convergência estrutural (mesmos motivos proteicos surgem de forma independente); e convergência de sequências (a ordem dos aminoácidos de proteínas ressurgem a partir de pressões evolutivas e não apenas ao acaso). Dos eventos de convergência molecular, o de convergência funcional (especialmente analogia enzimática), vem sendo identificado e proposto em vários estudos e continuamente novas buscas são feitas para caracterizar esse tipo de evento (23) (24), dada sua importância na compreensão dos fenômenos evolutivos relativos a proteínas e vias metabólicas. Por exemplo, a reconstrução de vias metabólicas em genomas completamente sequenciados utiliza comparação entre sequências para inferir quais genes e quais atividades enzimáticas compõem uma via metabólica específica, e apesar da eficiência desse método (25), genes sem função identificada representam entre 20% e 60% das proteínas na maioria dos genomas, criando uma profusão de *hypothetical proteins*¹. A discrepância entre genes esperados (que codificam enzimas que deveriam estar presentes na via metabólica) e os genes observados pode indicar que o organismo utiliza formas análogas para a atividade enzimática ausente (26).

Um estudo em busca por analogia enzimática foi feito em 1998 (27) onde foram identificados em 105 EC (de um total de 1.709) casos de enzimas que catalisam a mesma reação química, porém sem similaridade de sequências

¹ Proteína cuja existência é predita, porém, não há evidência sobre sua função ou mesmo sobre sua expressão.

significativa ou mesmo em motivos proteicos. Para 34 dos 105 EC foi demonstrado que pares candidatos a enzimas análogas possuíam formas estruturais muito distintas, não sugerindo qualquer relação evolutiva (27). Além desses indícios, buscas por analogia em sítios catalíticos enzimáticos sugerem que analogia enzimática não é um fenômeno raro (24).

O estudo de analogia enzimática pode contribuir no avanço do estudo em evolução e igualmente expor, através do estudo de seus processos catalíticos, alvos para o desenvolvimento de novas drogas. Isso se dá pelo fato de que muitas das enzimas análogas à forma enzimática do parasita são significativamente distintas da forma encontrada em seu hospedeiro o que pode indicar um potencial nicho de descoberta de novos fármacos (27), já havendo alguns estudos sistemáticos sobre o tema em *Trypanosoma cruzi* (28) e *Trichomonas vaginalis* (29) utilizando o princípio de busca por baixa similaridade estrutural entre enzimas que catalisam a mesma reação química em vias metabólicas fundamentais para esses parasitas.

Por fim, uma grande parte das atividades enzimáticas continua sem uma sequência proteica associada (30), atividades essas que podem ser chamadas de enzimas órfãs. Um estudo de 2014 (30) identificou mais de 1.000 atividades enzimáticas, dentre um total de 5.000, sem sequência relacionada, e muitas dessas não estão relacionadas a qualquer via metabólica conhecida. Um caso especial de enzimas órfãs é o de enzimas órfãs locais, atividades enzimáticas que não possuem uma sequência representativa em um clado específico, porém, possuem ao menos uma sequência em organismos pertencentes a outros clados, e nesse caso específico a existência de enzimas análogas pode ser a resposta (30). O estudo de analogia enzimática é, portanto, uma área que tem muito a contribuir na construção do conhecimento sobre o metabolismo e evolução de organismos, e igualmente contribuir no avanço da medicina em busca de novos alvos terapêuticos para doenças parasitárias.

Bancos de Dados Biológicos

Em Bioinformática são utilizadas classificações de bancos de dados não baseadas na tecnologia, mas sim pelo tipo de informação que armazenam. Tais bancos de dados biológicos somam mais de 1.552 (31), disponíveis para acesso

online, criados para diferentes propósitos, com diferentes níveis de cobertura e curados utilizando métodos diversos. Uma classificação possível para esses bancos de dados pode ser: primários e secundários (31). Bancos de dados primários são bancos cujos dados são armazenados sem qualquer tipo de processamento. Bancos de dados secundários, em contrapartida, possuem dados processados de alguma maneira, por exemplo, curados ou com informação adicional aos dados de bancos de dados primários.

Outra forma de classificar um banco de dados biológico pode ser como especializado e não especializado. Bancos de dados especializados possuem tipos específicos de dados ou relacionados a um grupo de organismos como, por exemplo, o WormBase (32) que possui dados exclusivamente sobre nematódeos. Bancos de dados não especializados armazenam tipos diferentes de informação e de diversos organismos. O GenBank (33) é um exemplo de banco de dados não específico que armazena dados de mais de 280.000 espécies de organismos. Uma classificação utilizada pela edição especial de banco de dados da revista *Nucleic Acids Research* (34) categoriza os bancos de dados de 15 diferentes maneiras: sequências nucleotídicas, sequências de RNA, sequências de proteínas, estrutura, genomas de invertebrados, metabolismo e sinalização, genomas de humanos e outros vertebrados, genes humanos e doenças, dados de *microarray* e outros dados de expressão gênica, recursos em proteômica, outros bancos de biologia molecular, organelas, plantas, imunologia, e por fim, biologia celular (35).

Permanência de bancos de dados

Conforme cresce a produção de dados biomédicos cresce também a necessidade de armazenar, compartilhar e organizar tais dados. O número de bancos de dados acessíveis pela Internet cresce anualmente e apesar do objetivo inicial de serem fontes importantes de informação, ao longo do tempo alguns destes se tornam inacessíveis, ou seja, são criados, mas não são mantidos (ou atualizados) e, ainda, alguns bancos de dados simplesmente nunca são utilizados. De todas as URLs publicadas em qualquer ano estima-se que em torno de 6% desaparecerão. Aproximadamente 20% das URLs publicadas em artigos MEDLINE (36) estão

inacessíveis e dessas URLs 20% são para bancos de dados. Alguns bancos de dados nunca são atualizados ou mantidos de forma eficiente e tais ações estão diretamente relacionadas à sua vida útil. Entre os bancos de dados existentes alguns são tão raramente acessados que podem ser caracterizados como “túmulos de dados” (37).

Kyoto Encyclopedia of Genes and Genomes (KEGG)

O projeto de desenvolvimento do KEGG foi iniciado em 1995 através do Programa Genoma Humano do Ministério de Educação, Ciência, Esportes e Cultura do Japão. O KEGG é uma base de conhecimento para análise sistemática de funções gênicas e outras informações correlatas (38) e é atualmente dividido em 17 bancos de dados, incluindo PATHWAY (vias metabólicas), GENES (catálogo de genes de genomas completamente sequenciados), REACTION (reações bioquímicas), entre outros (39). O KEGG fornece, dentre outros dados, sequências de genomas completamente sequenciados, dados sobre vias metabólicas e as relações entre organismos, atividades enzimáticas e seus mapas de redes metabólicas (38).

Structural Classification Of Proteins database (SCOP)

O SCOP é um banco de dados que fornece informação detalhada e abrangente das relações estruturais e evolutivas das proteínas com estrutura conhecida (40). A classificação das estruturas é feita em níveis hierárquicos (família, superfamília e enovelamento). As **famílias** indicam uma clara relação evolutiva, a identidade de pares de resíduos entre as proteínas é de 30% ou mais. Porém, em alguns casos, funções e estruturas similares evidenciam uma ancestralidade comum apesar de menor similaridade entre sequências. Por exemplo, muitas globinas formam uma família mesmo havendo 15% apenas de similaridade de sequência. Já as **superfamílias** indicam uma **provável** ancestralidade comum. São proteínas que

possuem baixa similaridade entre suas sequências, mas cujas funções e estruturas indicam uma mesma origem; ***foldi*** (enovelamento) indica apenas uma similaridade estrutural significativa. Proteínas são definidas como possuindo um enovelamento comum se possuírem a maioria das estruturas secundárias arranjadas da mesma maneira com as mesmas conexões topológicas.

Protein Data Bank (PDB)

O PDB é um banco de dados de estruturas tridimensionais de macromoléculas. Foi iniciado em 1971, no Laboratório Nacional de Brookhaven como um arquivo para estruturas cristalizadas de macromoléculas. Em princípio, o arquivo possuía um total de sete estruturas e a cada ano novas estruturas eram adicionadas. A partir da década de 1980 o número de estruturas depositadas começou a crescer, devido aos avanços na tecnologia de cristalografia, o surgimento da ressonância magnética nuclear como método para se obter estruturas de macromoléculas e as mudanças de visão da comunidade científica sobre compartilhamento de dados. No início da década de 1990 a maioria dos periódicos científicos passou a solicitar identificadores PDB para proteínas com estrutura 3D conhecida e ao menos uma agência de fomento (*National Institute of General Medical Sciences*) adotou as orientações da União Internacional de Cristalografia (IUCr) solicitando o depósito de todas as estruturas elucidadas no PDB (41).

Os dados disponíveis no PDB são armazenados em forma de arquivos contendo as coordenadas espaciais para todos os átomos de uma molécula. As moléculas depositadas no PDB são curadas e passam por um processo de validação até a sua disponibilização. Atualmente o PDB pode ser acessado via interface *web* (42) ou *API REST* (43).

A utilização dos dados do PDB é um passo importante em pesquisas com proteínas. A partir do PDB é possível obter arquivos descritivos sobre suas estruturas e imagens representativas de suas estruturas tridimensionais (Figura 2).

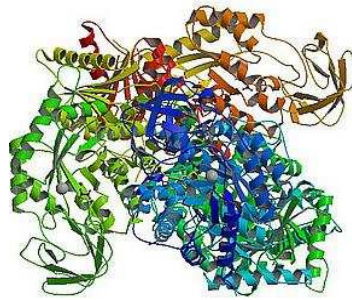


Figura 2. Representação da estrutura tridimensional da enzima piruvato cinase M2 obtida no *Protein Data Bank* (PDBid: 3BJT). As diferentes cores representam subunidades (ou cadeias) que formam a estrutura quaternária dessa proteína.

Bancos de Dados Relacionais

Em 1970 Edgar Frank Codd publicou um artigo que propunha a utilização de um modelo relacional de dados (44). Nesse documento, dentre outros apontamentos, o esquema de banco de dados e sua organização lógica se apresentavam de forma desconectada do armazenamento físico dos dados, e os dados e suas relações eram representados através tuplas. Ao longo da década de 70 diversos sistemas de bancos de dados foram criados utilizando esse modelo, cujas evoluções deram origem também à linguagem SQL (utilizada para manipular os dados desses sistemas) e nos diferentes sistemas de banco de dados relacionais disponíveis atualmente (ORACLE, PostgreSQL, MS SQL Server, MySQL etc). As estruturas de dados num banco de dados relacional são representadas por tabelas (constituídas por linhas e colunas) que podem ser relacionadas a outras tabelas como demonstrado no exemplo da Figura 3.

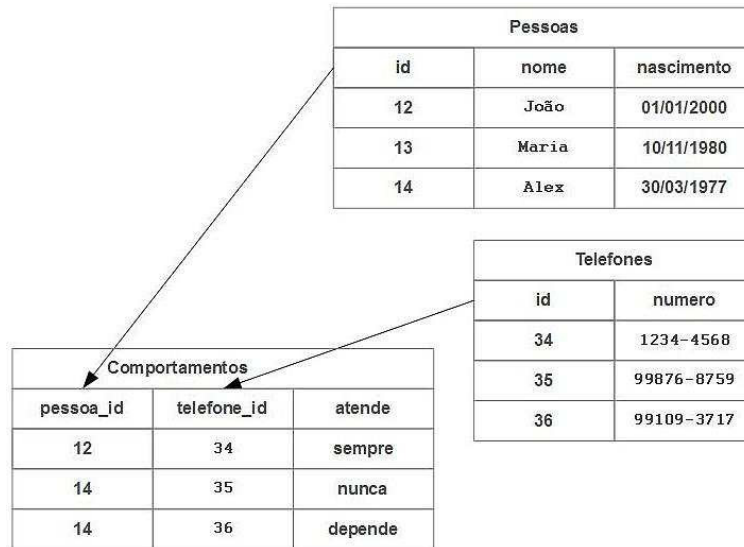


Figura 3. Demonstração gráfica de organização de tabelas em um banco de dados relacional. Duas tabelas (Pessoas e Telefones) são relacionadas por uma terceira tabela a partir das colunas *id*, *pessoa_id* e *telefone_id*.

Metodologia de desenvolvimento de *software*

Uma metodologia de desenvolvimento de *software* é uma estrutura (*framework*) de conceitos utilizados para estruturar, planejar e controlar o processo de desenvolvimento de um sistema (45). Sistemas minimamente complexos exigem que um controle formal atue sobre o processo de desenvolvimento a fim de minimizar riscos e defeitos no *software* e maximizar a produtividade.

Ao longo do tempo diversas metodologias (ou abordagens dentro de metodologias) foram desenvolvidas para diferentes finalidades, tamanhos de equipe de desenvolvedores e demandas de tempo e custo. Não é possível listar todas as metodologias uma vez que continuam surgindo novas metodologias ao longo do tempo. Além disso, uma categorização das metodologias não é unanimidade dentro da engenharia de *software*, assim como não é unanimidade se alguns nomes de metodologia se referem a um conjunto finito e fechado de princípios ou se são apenas um grupo de visões possíveis para serem adotadas por equipes de desenvolvimento de *software*. A abordagem mais comum é dividir as metodologias entre **tradicionais** e **ágeis**, onde metodologias tradicionais são caracterizadas por

um forte planejamento desde o início até a entrega do software, por serem rigorosamente baseadas em documentação e por admitirem com maior dificuldade modificações ao longo do projeto. Metodologias ágeis são normalmente caracterizadas por priorizarem a entrega imediata de partes funcionais do *software*, considerarem modificações no projeto como inerentes a projetos de *software* e pela interação ininterrupta e necessária entre todos os envolvidos no projeto (clientes, desenvolvedores e gestores).

OBJETIVOS

Objetivo Geral

Desenvolvimento de um sistema *web* de acesso público (AnEnDB) contendo uma nova versão da ferramenta para a predição computacional de enzimas análogas (AnEnPi-v2) e de um banco de dados de sequência, estrutura e evolução de enzimas análogas.

Objetivos Específicos

1) Implementar e melhorar a metodologia para agrupamento de sequências na busca e identificação de analogia (AnEnPi-v2);

2) Construir um banco de dados a partir dos dados oriundos do AnEnPi-v2 que permita análises de sequência, estrutura e evolução de enzimas análogas nos três domínios da vida;

3) Implementar a ferramenta de reconstrução metabólica disponibilizada pelo KEGG para mapear as enzimas análogas;

4) Construir uma interface gráfica *web* que permita a utilização da ferramenta AnEnP-v2 e a análise dos dados depositados no banco de dados.

MATERIAL E MÉTODOS

Metodologia de desenvolvimento de *software*

O AnEnDB foi desenvolvido utilizando uma adaptação de elementos de metodologias Ágeis como *Scrum* (46) e XP (47) por conta das características continuamente mutáveis do projeto, especialmente a utilização de um *backlog* (quadro de tarefas como na Figura 4), propriedade coletiva do código e entrega contínua de partes funcionais do *software*.














PBI	TODO	<u>In Progress</u>	<u>Done</u>
			
			
			
			

Figura 4. *Product Backlog*. Ferramenta utilizada em *Scrum* para acompanhamento de tarefas. PBI é a coluna *Product Backlog Item* e é onde são listados, em linhas gerais, os recursos a serem desenvolvidos. A coluna *TODO* (contração do inglês *to do*) é a coluna de tarefas específicas de cada recurso. A coluna *In Progress* possui as tarefas selecionadas e que estão de fato sendo implementadas. A coluna *Done* possui as tarefas que já foram concluídas.

Divisão do projeto em áreas de estudo e atuação

O projeto do AnEnDB é dividido em seis elementos, como exemplificado na Figura 5:

- i) origem de dados (KEGG)
- ii) extratores de dados (nomeados *parsers*)
- iii) algoritmo de agrupamento de enzimas análogas
- iv) banco de dados relacional
- v) ambiente de desenvolvimento
- vi) apresentação de dados

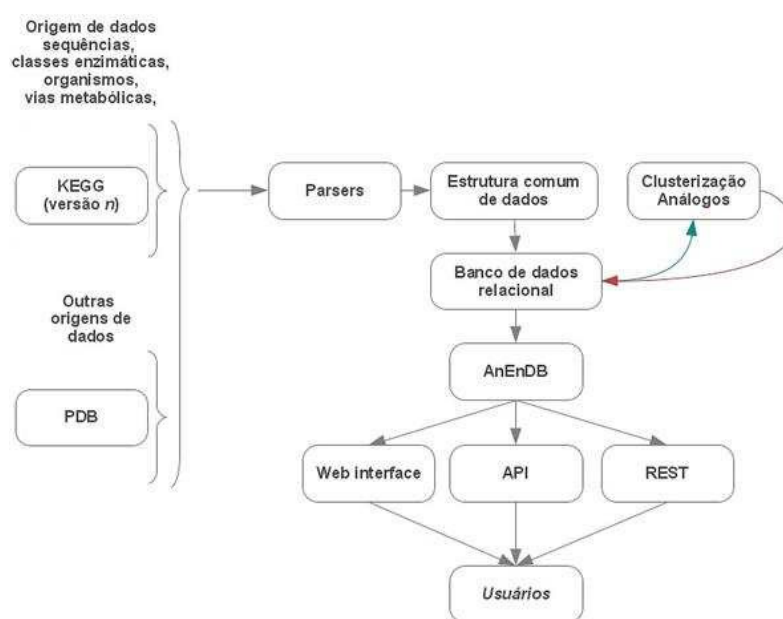


Figura 5. Desenho experimental do AnEnDB. Os componentes *Parsers*, *Banco de dados relacional*, *Clusterização Análogos* e *AnEnDB*, constituem os elementos mais importantes e representam o ambiente funcional do sistema.

Origem de dados

Para a atual versão do AnEnDB está sendo utilizado o KEGG, versão de fevereiro de 2015.

Repositórios dessa natureza normalmente reúnem as informações em arquivos texto ASCII. Apesar de esses arquivos estarem relacionados através de algum tipo de identificador (código de acesso da proteína, classe de atividade enzimática, entre outros), cada arquivo possui sua própria lista de dados e informações, ou seja, muitos desses arquivos possuem dados redundantes, dispersos e muitas vezes seus relacionamentos são definidos apenas entre nomes de arquivos. Por exemplo, a relação entre classes de atividades enzimáticas e organismos está em arquivos no formato *código_do_organismo_enzyme.list*. O arquivo *mid_enzyme.list* por exemplo, que relaciona as proteínas do organismo *Mycobacterium indicus pranii* com classes de atividades enzimáticas possui 13 registros para a mesma atividade enzimática 2.7.7.7 (Figura 6):

```
mid:MIP_00002    ec:2.7.7.7
mid:MIP_00467    ec:2.7.7.7
mid:MIP_00703    ec:2.7.7.7
mid:MIP_00718    ec:2.7.7.7
mid:MIP_00839    ec:2.7.7.7
mid:MIP_02583    ec:2.7.7.7
mid:MIP_03183    ec:2.7.7.7
mid:MIP_04428    ec:2.7.7.7
mid:MIP_04535    ec:2.7.7.7
mid:MIP_04549    ec:2.7.7.7
mid:MIP_05944    ec:2.7.7.7
mid:MIP_06331    ec:2.7.7.7
```

Figura 6. Parte do conteúdo do arquivo *mid_enzyme.list* do KEGG. O arquivo possui a lista de identificadores de proteínas (coluna da esquerda) para a atividade enzimática 2.7.7.7 (coluna da direita) do organismo *Mycobacterium indicus pranii*.

O arquivo que possui as sequências proteicas do mesmo organismo estão localizados em outro diretório cujo arquivo é nomeado como *T02325.pep* (Figura 7):

```

>mid:MIP_00467 DNA polymerase IV 2 (EC: 2.7.7.7)
MFVRC DPSILHADLDSFYASVEQRDDPALRGRPVIVGGGVVLAASYEAKAYGVRTA...
>mid:MIP_00470 hypothetical protein
MTTAERLLQERPLADISVDDLAKGAGLSRPTFYFYFPSKDAVLFTLFEVIMEAAA...
>mid:MIP_00469 transcriptional regulatory repressor protein
MASSRSSPDGRAERLPRWVRAEPTVVTAATLHGVEIFDITVLT DSTPC
>mid:MIP_00471 Monooxygenase, flavin-binding family protein
MTEHLDVLIIVGAGISGVSAAWHLQERCPTKSYAILE RRADLGGTWDLFKYPGIRSDSDF
HPGNDVDELPMDFTPGYFRRSMHLLPKSGSRAPWRLKQNYFFDMRTIRRGKVDDEG...
AKKPAPVAV

```

Figura 7. Parte do conteúdo do arquivo T02325.pep, do KEGG, que possui as sequências proteicas do organismo *Mycobacterium indicus pranii*.

Igualmente, arquivos sobre vias metabólicas e identificadores PDB estão localizados em outros diretórios e arquivos.

Para indexar e relacionar as informações do banco de dados KEGG é necessário um pré-processamento dessas informações, através de uma camada de *software*, especializada em processamento de texto para filtrar, cortar, unir, separar e tornar consistente as informações do banco de dados primário com o tipo de estrutura de dados que o AnEnDB utiliza. Por exemplo, o AnEnDB precisa transpor os dados do KEGG para a estrutura de dados do tipo *protein*, ou seja, estruturas que contenham o identificador da proteína, suas classes de atividade enzimática, organismo relacionado, entre outros, sem qualquer duplicidade, ausência ou inconsistência.

O AnEnDB possui classes específicas (Figura 8) para esse tipo de processamento de texto que fazem uso de expressões regulares (48) e algoritmos próprios. Tais classes fazem parte de uma camada dedicada a esse tipo de operação e pode ser modificada para agregar novos bancos de dados primários sem afetar as camadas superiores.

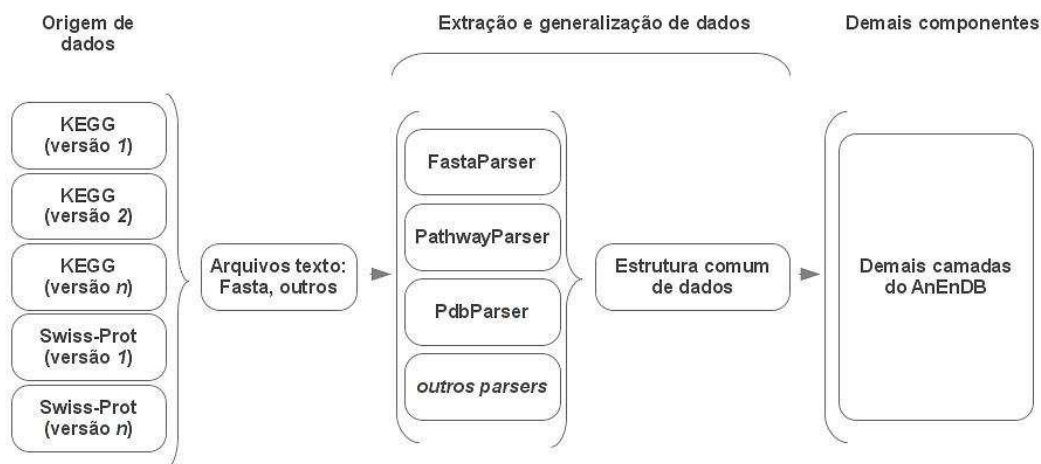


Figura 8: Diagrama demonstrando a relação entre camada de *extratores de dados*, origem de dados e apresentação dos dados.

Algoritmo de agrupamento de enzimas análogas

O agrupamento de enzimas análogas foi feito utilizando o mesmo algoritmo desenvolvido pelo AnEnPi (49). Nesse algoritmo, proteínas de todas as atividades enzimáticas identificadas em genomas completamente sequenciados (provenientes do KEGG) são agrupadas por atividade enzimática, com base no grau de similaridade entre suas estruturas primárias, de acordo com a metodologia descrita por Otto e colaboradores (2008), fundamentada em premissas estabelecidas em um estudo anterior de Galperin e colaboradores (27). Resumidamente, as sequências proteicas são primeiramente separadas por suas classes de atividades enzimáticas (determinadas por seus *EC numbers*); o passo seguinte consiste em remover todas as sequências que possuem menos de 100 aminoácidos para evitar a presença de possíveis artefatos. Em seguida, é executado um algoritmo de busca por similaridade que utiliza alinhamento local (Basic Local Alignment Search Tool - BLAST) (50) em que todas as sequências são comparadas contra todas as sequências pertencentes à mesma atividade enzimática. As sequências cujas comparações resultem em um *score* maior do que 120 são consideradas similares o bastante para serem consideradas homólogas e são agrupadas (18), enquanto

aquelas cujo resultado é menor do que 120 são consideradas análogas e se aloca em grupos distintos, ou seja, não possuem similaridade significativa entre suas sequências de aminoácidos (e provavelmente entre suas estruturas terciárias, o que deve ser verificado em uma etapa posterior), porém compartilham a mesma atividade enzimática. O resultado final são grupos (*clusters*) de sequências homólogas, formados dentro de cada atividade enzimática que, entre si, são análogos.

O algoritmo do AnEnDB que executa o agrupamento, representado na Figura 9, foi reproduzido integralmente a partir do AnEnPi. Sua única diferença é em relação à linguagem de programação utilizada. O AnEnPi utilizava a linguagem Perl (51) enquanto o AnEnDB utiliza a linguagem Python (52).

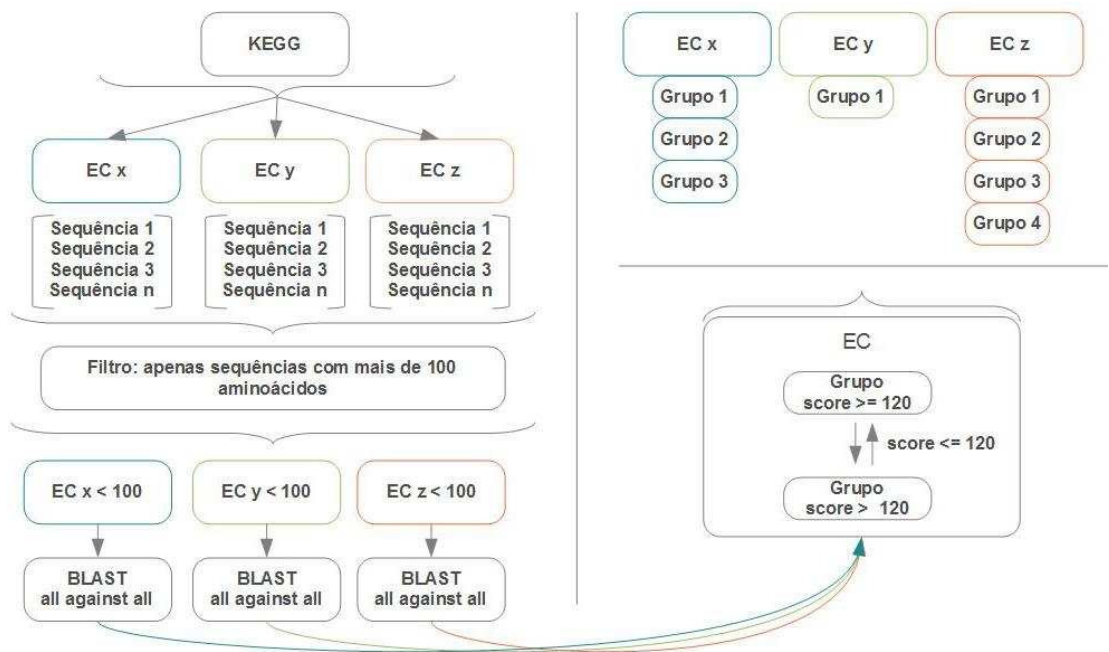


Figura 9. Diagrama demonstrando a metodologia de agrupamento do AnEnDB. A partir de bancos de dados como o KEGG as sequências são separadas por classes de atividade enzimática (números EC). Em seguida passam por um filtro para remover sequências com menos de 100 aminoácidos. Cada grupo de sequências (por número EC), passa pela execução de um algoritmo de busca por similaridade usando a abordagem todos contra todos. O resultado são grupos de sequências homólogas que, entre si, são análogos.

Banco de Dados Relacional

O AnEnDB utiliza o SGBD PostgreSQL (53). O PostgreSQL é um SGBD relacional criado há mais de 15 anos (e em contínuo desenvolvimento) de código aberto (54) e distribuído sob a licença *PostgreSQL Licence* (55). Oferece suporte a diversos sistemas operacionais (Linux, UNIX, AIX, SGI, Mac OS X, Windows, entre outros), acesso multiusuário, transações e suporte a diversos tipos de dados mais utilizados (varchar, integer, boolean etc).

Ambiente de Desenvolvimento

Existe uma grande variedade de linguagens de programação disponíveis para desenvolver sistemas. O AnEnDB é um *software* principalmente voltado para a apresentação de dados utilizando protocolos da Internet, mais especificamente *Hypertext Transfer Protocol* (HTTP) (56). Apesar da opção por um *software web* indicar um conjunto mais reduzido de linguagens ideais para o desenvolvimento de sistemas, é possível desenvolver *softwares web* em praticamente qualquer linguagem de programação. Estabeleceram-se então alguns critérios para a escolha da linguagem, de acordo com o contexto do projeto: tempo máximo de 12 meses para finalização, disponibilidade de apenas um programador e igualmente responsável por todos os aspectos de desenvolvimento do sistema, ambiente acadêmico (em termos de financiamentos e cultura tecnológica) e fluência específica do único desenvolvedor do projeto nas diferentes linguagens disponíveis.

Linguagens de programação se tornam mais ou menos populares em diferentes meios ao longo do tempo e não há consenso em torno de um único método para avaliar essa característica. Apesar disso, a popularidade é um dado importante pois impacta diretamente na vida útil de um *software*. *Softwares* desenvolvidos em linguagens pouco utilizadas exigem profissionais que se tornam escassos e caros para dar continuidade ao seu desenvolvimento (atualizações e manutenção).

Para analisar a popularidade de linguagens foram utilizados os sistemas GitHub (57), Google Trends (58), Tiobe (59) e Redmonk (60). GitHub, Tiobe e Redmonk indicam no ano de 2016 as 22 linguagens mais populares: JavaScript,

Java, Python, CSS, PHP, Ruby, C++, C, Shell, C#, Objective-C, R, VimL, Go, Perl, VisualBasic .Net, Visual Basic, Assembly Language, Delphi, MATLAB, Swift e Scala.

Os critérios relacionados ao tempo máximo de 12 meses de execução e financiamento para o AnEnDB impuseram a remoção das linguagens, Java, C++, C, C#, Objective-C, VisualBasic .Net, Visual Basic, Assembly e MATLAB. Tais linguagens ou exigem uma elaboração mais sofisticada e conseqüentemente maior gasto de tempo, ou simplesmente são proprietárias. O critério de fluência do único desenvolvedor do AnEnDB retirou da lista as linguagens Swift, VimL, Go, Delphi e Scala. Restaram as linguagens JavaScript, Perl, Python, CSS, PHP, Ruby e Shell. Por não serem linguagens para desenvolver sistemas e sim linguagens auxiliares (embora utilizadas extensivamente no AnEnDB pois são linguagens para lidar com aspectos gráficos da apresentação da interface), foram excluídas JavaScript e CSS. A linguagem Shell foi excluída devido à ausência de recursos adequados para desenvolvimento *web* (embora seja possível com o esforço necessário; mas recairia no critério de tempo máximo de 12 meses).

A lista foi, portanto, reduzida às linguagens Python, Perl, PHP e Ruby. Para analisar juntamente o critério de popularidade da linguagem somado à cultura tecnológica do meio acadêmico utilizamos como fonte o estudo conduzido por Marcia Chappel (61) que buscou identificar as linguagens mais utilizadas nas universidades listadas como as 10 melhores pela revista Forbes em 2013. Ao considerar apenas as quatro linguagens separadas anteriormente (Perl, Python, PHP e Ruby) a linguagem Ruby sequer aparece na lista das 20 mais utilizadas. Python, Perl e PHP aparecem na ordem de mais utilizadas como 4^a, 11^a, e 12^a, respectivamente. Ao selecionar deste grupo Python e Perl como possíveis linguagens para desenvolver o AnEnDB foi feita uma busca por popularidade no Google Trends (Figura 10 e Figura 11) pelas palavras “python” e “perl”.



Figura 10. Comparação entre as buscas por “Python” (curva azul) e “Perl” (curva vermelha) no mundo a partir do Google Trends. Fonte: <http://trend.google.com>.

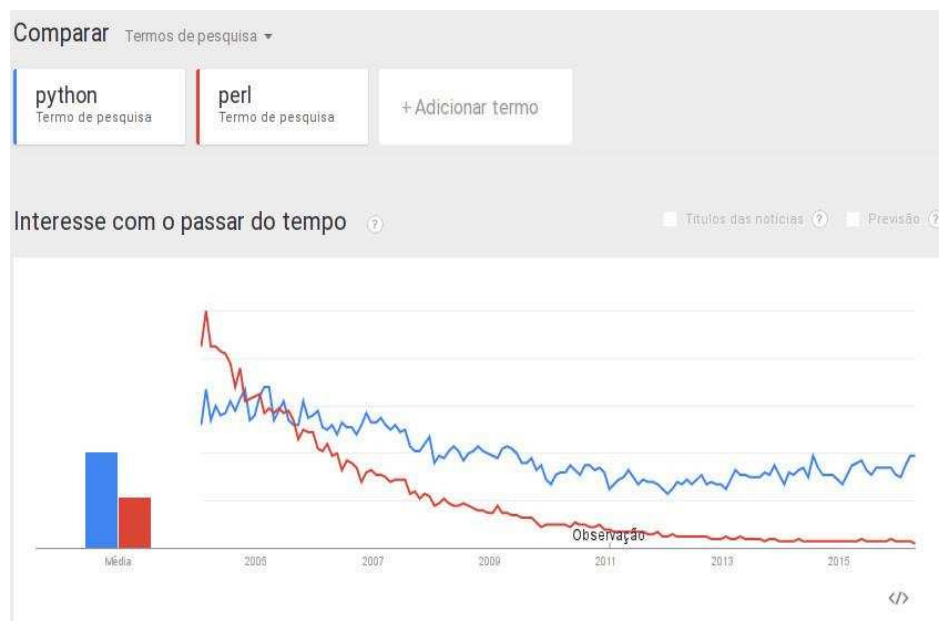


Figura 11: Comparação entre as buscas por “Python” (curva azul) e “Perl” (curva vermelha) no Brasil a partir do Google Trends. Fonte: <http://trend.google.com>.

A partir da análise dos dados sobre popularidade de linguagens de programação e ponderados os critérios para o desenvolvimento do sistema, foram considerados ainda elementos técnicos essenciais: orientação a objetos nativa,

suporte a módulos e *plugins* auxiliares etc. Após confirmar que ambas as linguagens (Python e Perl) oferecem todos os recursos técnicos necessários, ou seja, não havia limitações técnicas para uma escolha, a linguagem Python foi escolhida por estar de acordo com todas as necessidades do AnEnDB e estar representada como uma linguagem que cresce em popularidade há aproximadamente 10 anos.

A decisão sobre a linguagem é importante, pois aponta escolhas subsequentes sobre o ambiente de desenvolvimento a ser criado (fornecedores de plug-ins e módulos auxiliares, também se partes do software precisarão de componentes proprietários, disponibilidade de documentação de ajuda, suporte aos bancos de dados que serão utilizados, dentre outros elementos). Os elementos mais importantes do ambiente de desenvolvimento do AnEnDB são: módulo para abstração de banco de dados, ferramenta para documentação de código e um *framework* para desenvolvimento web.

Módulo para abstração de banco de dados

Object/Relational Mapping (ORM) é uma técnica que permite ao programador manipular dados de um banco de dados relacional utilizando o paradigma de orientação a objetos. O módulo ORM utilizado pelo AnEnDB é o SQLAlchemy (62). A utilização de ORM (Figura 13) no AnEnDB serve para tornar mais abstrato o acesso aos dados do banco de dados relacional e tornar mais rápido o desenvolvimento do *software*.

No primeiro caso significa principalmente separar a tecnologia, fornecedor e versão do banco de dados relacional das estruturas de dados utilizadas pelo sistema (AnEnDB), em outras palavras, o AnEnDB utiliza as tabelas do banco de dados como objetos e fica a cargo do módulo ORM lidar com a manipulação dos dados diretamente do banco de dados relacional, seja ele MySQL, PostgreSQL, ORACLE, etc. Dessa maneira não é necessário escrever código em linguagem SQL. Igualmente não é necessário conhecer detalhes de implementação do banco de dados referentes ao fornecedor do *software*. Ainda, um módulo ORM oferece meios de acessar as estruturas de dados de maneira já relacionada, ou seja, se o AnEnDB possui uma entidade *Protein*, então ao acessar *Protein* é possível ler diretamente suas classes de atividade enzimática, organismo, mapas de vias metabólicas etc,

sem ter que escrever código específico para mapear todas essas relações, muito menos ter que mapear os nomes de campos em resultados de *queries* para variáveis.

No segundo caso o ganho de velocidade no desenvolvimento de *software* se dá na eliminação de codificação redundante: o uso de instruções SQL (Figura 12) diretamente impõe a necessidade de reescrever os comandos SQL (INSERT, SELECT, UPDATE, etc) a cada manipulação dos dados, conhecer a estrutura do banco de dados (é necessário saber, por exemplo, como proteínas e mapas de vias metabólicas são relacionados em termos de índices de tabelas) e por fim auxilia o programador para assegurar que o *software* não sofra com, por exemplo, *SQL injections* (técnica que permite usuários inserirem códigos maliciosos via URL do navegador de internet e que afetam diretamente a consistência do banco de dados relacional).

```
#!/usr/bin/python
db = connect( 'mysql://user:password@localhost/anendb' )
db.execute( "INSERT INTO \
              protein( identification, sequence ) \
VALUES ( \
          'mma:T00023', \
          'MGYKCTRCKQKVEIDYEYTGIRCPYCGHRILVKERPTTIKRIKAE' \
        );
```

Figura 12. Exemplo de script Python para efetuar uma operação em um banco de dados relacional (sem a utilização de ORM). O código demonstra que para inserção de um registro no banco de dados é necessária a codificação direta de instruções SQL (INSERT, por exemplo).

```
protein = Protein()
protein.identification = ' mma:T00023'
protein.sequence      = 'MGYKCTRCKQKVEIDYEYTGIRCPYCGHRILVKERPTTIKRIKAE'

session.add(protein)
session.commit()
```

Figura 13. Exemplo de script Python para efetuar uma operação em um banco de dados relacional (com a utilização de ORM). O código demonstra que as tabelas do banco de dados são referenciadas através de objetos (Protein, como no exemplo) e suas propriedades (como demonstrado na propriedade *sequence*). A ORM lida com aspectos internos do banco de dados (os nomes reais das tabelas etc) o que torna o código mais reutilizável.

A utilização de ORM em desenvolvimento de sistemas não é unanimidade (63) e pôde ser verificada na prática ao longo do desenvolvimento do AnEnDB. Enquanto operações de busca e atualização de dados foram simplificadas (sem a necessidade de escrever código SQL), um tempo de execução excessivo para o agrupamento de enzimas (grupos de análogos) foi adicionado. Por essa razão a utilização de ORM, apesar de extensamente utilizada, é restrita aos aspectos do AnEnDB que se beneficiam da simplificação e clareza do código. Operações computacionalmente mais demoradas, como por exemplo, a identificação, verificação e inserção de milhões de registros, e afetadas pelo desempenho de ORM, utilizaram a geração de arquivos texto ASCII com código em linguagem SQL para serem executados manualmente pelo desenvolvedor.

Ferramenta para documentação de código

A documentação é uma parte fundamental do desenvolvimento de um *software*, pois permite: esclarecer os objetivos do projeto, requisitos e atividades; desenhar e especificar o *software*; tornar o *software* fácil de entender; permitir que outros programadores possam trabalhar no código já escrito; auxiliar na comunicação correta entre usuários e demais envolvidos no *software* (64).

Existem vários tipos de documentação possíveis de acordo com a metodologia escolhida e especificações do *software*. O AnEnDB possui documentação para os seguintes aspectos:

Arquitetura/desenho

São diagramas que demonstram alguns princípios de construção do *software*, componentes e suas relações, entre outros aspectos da arquitetura/desenho; esse tipo de documentação atende aos interessados em conhecer como o AnEnDB foi modelado em linhas gerais.

Tal documentação é importante para compreender e identificar como as partes que compõem o AnEnDB lidam com a demanda de poder computacional e volume de dados e como segmenta as diferentes camadas do projeto.

Técnica

A documentação técnica disponível pelo AnEnDB se restringe à documentação sobre os métodos e classes dos sistema. Ela é em parte gerada automaticamente através do *software* Sphinx (65) e em outra parte gerada manualmente (documentação sobre uso da API do AnEnDB – quais métodos estão disponíveis, exemplos de como usá-los e resultados esperados). O Sphinx obtém automaticamente, a partir do código fonte do *software* e comentários especiais, nomes de classes, métodos e classes do código do AnEnDB e produz um conjunto de páginas HTML disponíveis para leitura. Esse tipo de documentação não fica disponível para os usuários finais e são exclusivas para desenvolvedores do *software*.

Com essa documentação um programador pode visualizar rapidamente quais métodos estão disponíveis no AnEnDB e suas funções. Isso permite que o programador encontre rapidamente o que precisa para atualizar o *software* ou simplesmente garantir que ele não escreva código redundante.

O Sphinx gera um sistema de busca para métodos e classes e demais elementos adicionados pelo desenvolvedor. A partir da leitura da documentação técnica um desenvolvedor é capaz de identificar onde deve criar código de atualização e onde deve criar código de otimização.

Usuário final

A documentação para o usuário final é a que demonstra como utilizar o AnEnDB na prática, seus princípios metodológicos e exemplos de resultados. A documentação para o usuário final pode ser acessada no *link documentação* da interface web do AnEnDB.

Framework para desenvolvimento web.

Existem várias definições para *framework*. Para o AnEnDB, em termos de ambiente de desenvolvimento, a definição utilizada é a de que um *framework* é um conjunto de classes que agrega um design abstrato de soluções para uma família de problemas correlatos (66). Na prática significa que o AnEnDB utiliza um *framework* de desenvolvimento como um conjunto de recursos de *software* prontos que podem ser utilizados sem a necessidade de conhecer ou implementar os detalhes desses recursos. Por se tratar de um *software* voltado para a *web* e principalmente escrito em Python, o AnEnDB utiliza o *framework* Flask (67). Flask é um *framework* para Python baseado no Werkzeug (68) e Jinja 2 (69).

Werkzeug é uma biblioteca Python utilizada para aplicações Web Server Gateway Interface (WSGI) (70). WSGI é uma especificação que descreve como um servidor web se comunica com aplicações web e como aplicações web podem ser encadeadas para processar uma requisição. A comunicação entre servidores web e aplicações web se beneficia de padronizações, pois aplicações web não precisam lidar com detalhes do protocolo HTTP ou mesmo com implementações internas de cada servidor HTTP (Apache, Nginx etc); em outras palavras, aplicações web podem ser codificadas de maneira transparente às implementações internas do protocolo HTTP e dos diferentes tipos de servidores HTTP existentes.

Além de Werkzeug, Flask implementa Jinja 2, uma importante biblioteca que facilita a relação entre os dados gerados pelo *software* e sua apresentação. Numa aplicação web típica, baseada em apresentação de dados via páginas formatadas na linguagem HTML, existe a necessidade de apresentar estruturas de dados dinâmicas dentro de formatos rígidos (HTML). Jinja 2 viabiliza a utilização de linguagem dinâmica (Python) dentro de linguagens estáticas (HTML por exemplo), como demonstrado na Figura 14.

```
<html><head><title>AnEnDB</title></head><body>
  {% for organism in organisms %}
    {{ organism }} <br />
  {% endfor %}
</body></html>
```

Figura 14. Exemplo de código de página HTML que lista todos os organismos do AnEnDB. O código utilizado precisa apenas acessar a lista `organisms` e iterar sobre ela, sem a necessidade de qualquer conhecimento sobre como a lista `organisms` é gerada.

Flask e sua implementação WSGI juntamente com Jinja 2 fornecem um ambiente de desenvolvimento que permite ao AnEnDB apresentar os dados sem ter que lidar com detalhes de implementação do protocolo HTTP e sem ter que lidar com as limitações de linguagens de formatação e estruturação estáticas (HTML, XML etc).

Apresentação de dados

A apresentação dos dados do AnEnDB é feita de duas maneiras distintas: interface *web* e REpresentational State Transfer (REST) (71).

Interface web

A interface web do AnEnDB é um conjunto de páginas HTML (72) geradas dinamicamente através da interação entre usuário, módulo `mod_wsgi` (WSGI) e servidor HTTP, representados nas Figura 16 e Figura 15.

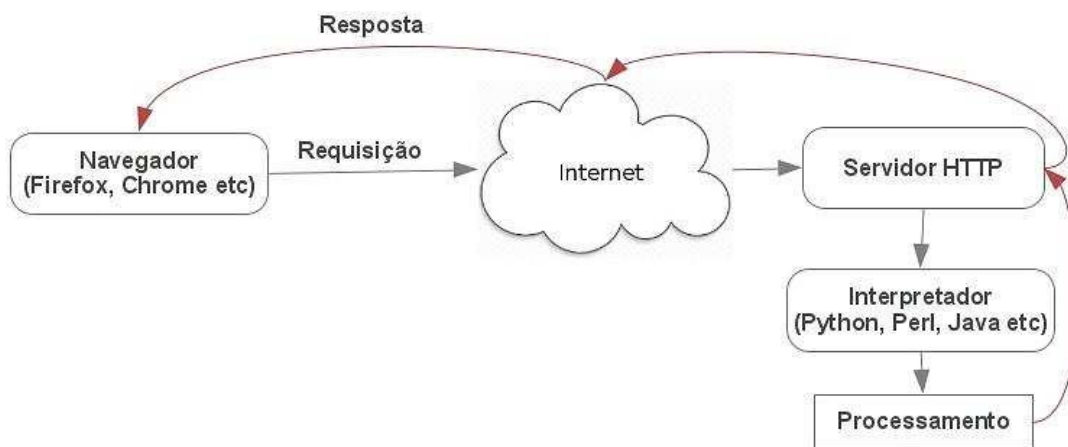


Figura 16. Representação do fluxo de uma página web dinâmica. O usuário, a partir de um navegador, solicita um recurso ao servidor HTTP. O servidor HTTP repassa a solicitação para o interpretador da linguagem de programação que processa o conteúdo do recurso e devolve (linhas vermelhas na figura) para o navegador através da internet.

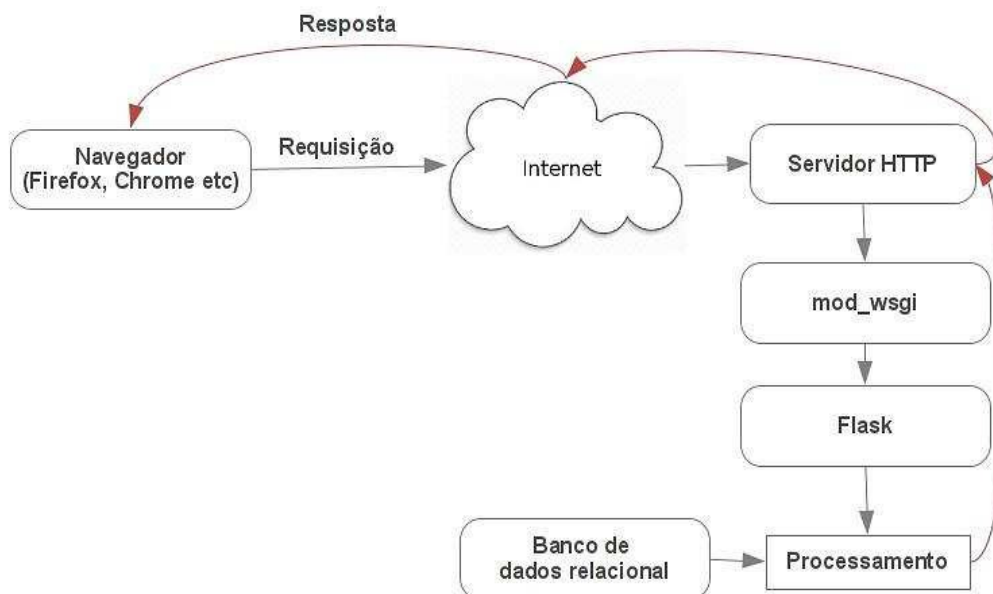


Figura 15. Representação do fluxo de uma página web dinâmica utilizando o framework Flask e WSGI. O usuário, através de um navegador, solicita um recurso ao servidor HTTP. O servidor HTTP repassa a solicitação para a o módulo mod_wsgi e em seguida para o framework Flask que executa o processamento e retorna o resultado para o usuário.

REpresentational State Transfer (REST)

REST é um estilo de arquitetura e uma abordagem que é frequentemente usada em serviços web. A arquitetura desacoplada de REST e comunicação mais simples e leve entre requisitante e fornecedor torna REST um estilo popular para construir aplicações web. Quando serviços web utilizam REST como arquitetura estes são chamados de APIs RESTful ou APIs REST.

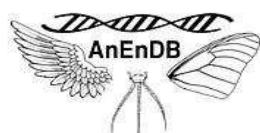
O AnEnDB utiliza REST através do protocolo HTTP e sua arquitetura REST é implementada pelo *framework* Flask. A utilização de REST no AnEnDB liberta o usuário de ter que interagir com um navegador web ou utilizar recursos gráficos de interface para obter resultados sobre analogia de enzimas. O usuário pode, por exemplo, criar scripts próprios em qualquer linguagem de programação que retorne os resultados que deseja.

RESULTADOS E DISCUSSÃO

Acesso ao AnEnDB

O AnEnDB pode ser acessado via navegador *web* no endereço (Figura 17):

<http://157.86.220.224:5000/>



[Home](#) [About](#) [Documentation](#) [Download](#) [References](#) [Methodology](#) [Contact](#) [Search](#)

Search analogous by:

- [Intergenomic search by organism](#)
- [Intragenomic search by organism](#)
- [Intragenomic Analogy by EC Number](#)
- [EC Number](#)

AnEnDB is a software for analogous enzyme searching.

Analogous enzymes are enzymes that has the same biochemical function but with significant structural differences.

In other words, are enzymes that evolved independently to reach the same biological function.

Seems that life, sometimes, doesn't matter the way in the evolution, finds a way out to fulfill its needs.

Birds and insects finds theirs ways to get wings to fly, for example.

Same way, internal cell biochemical functions - enzymes - finds their ways to keep life running as just as it's necessary.

AnEnDB aims to help researchers to clarify the analogous enzymes creation mechanisms and its secondary outspreads.

References:

[AnEnPi: identification and annotation of analogous enzymes.](#)

Figura 17. Primeira página (Home) da interface web do AnEnDB.

Metodologia de Desenvolvimento de Software

Métodos tradicionais de desenvolvimento de *software* são fortemente prescritivos e se caracterizam pelo foco em planos detalhados e definidos no princípio do projeto. Possuem custo, escopo, cronograma detalhado, microgerenciamento, poder centralizado, processos cada vez mais complicados e extensa documentação. Mudanças são fortemente indesejadas. Metodologias tradicionais acreditam que seria possível tratar o desenvolvimento de *software* como um processo previsível. O método tradicional mais conhecido para gerenciamento de projetos é o modelo *Waterfall*, inicialmente descrito por *Royce* em 1970. *Royce* no entanto criticava o modelo em seu artigo afirmando que para desenvolvimento de *software* seu uso era arriscado (46).

Em 1990 já eram descritos os motivos pelos quais os métodos tradicionais de desenvolvimento de *software* não funcionam a partir das prerrogativas usuais: requisitos não são completamente compreendidos antes do início do projeto; usuários só sabem exatamente o que querem após ver uma versão inicial do produto; requisitos mudam frequentemente durante o processo de desenvolvimento e novas ferramentas e tecnologias tornam as estratégias de desenvolvimento imprevisíveis (46).

Para definir a metodologia de desenvolvimento adequada para o AnEnDB as seguintes características foram levadas em consideração:

- i) os “clientes” ao longo do desenvolvimento seriam os orientadores do projeto;
- ii) apenas um indivíduo codificaria todo o *software* até a primeira versão em produção;
- iii) os custos, assim como qualquer financiamento, seriam imutáveis;
- iv) o prazo para entrega do *software* seria fixo (máximo de 12 meses);
- v) o *software* poderia passar por diversas modificações estruturais ao longo do desenvolvimento;
- vi) o conhecimento sobre o banco de dados primário seria desenvolvido ao longo do processo;

vii) os “clientes” não dariam *feedback* em intervalos fixos e sim de acordo com suas disponibilidades;

viii) não haveria funções de gerência de projeto exclusivas para validar documentações;

ix) não haveria funções técnicas exclusivas para testar protótipos;

x) não haveria procedimento formal de aceite para cada etapa desenvolvida.

Dadas as características listadas, alguns elementos fundamentais de algumas metodologias teriam que ser adaptados:

i) não haveria prototipagem a cada etapa;

ii) não haveria programação em pares;

iii) não seria possível planejar todo o *software* antes de iniciar a codificação;

iv) manter atualizadas documentações completas (em diversos formatos como, por exemplo, UML) não geraria valor relevante dado o prazo para o desenvolvimento.

Sem prototipagem, sem planejamento completo antes da codificação e sem equipe disponível para trabalhar exclusivamente em documentações e coordenação entre “cliente”, equipe e produto, o AnEnDB teve que ser desenvolvido adaptando elementos de metodologias ágeis como Scrum e XP.

De Scrum, as funções de *Product Owner* e *Scrum Master* foram assumidas pelo desenvolvedor e orientadores. De metodologias ágeis de um modo geral foram adotadas as características de utilizar o próprio código como documentação relevante, testes de unidade para manter a consistência dos requisitos e a aceitação de mudanças contínuas ao longo do projeto.

Um *product backlog* nos modelos definidos em Scrum, com lista de itens organizados por prioridade, foi utilizado para garantir que os itens de maior valor fossem desenvolvidos primeiro. O objetivo foi evitar desenvolvimentos desnecessários e, principalmente, garantir um conjunto de recursos prontos que pudessem ser, de fato, entregues funcionalmente no fim do prazo.

De XP foram utilizados os princípios de padronização de código e propriedade coletiva. A padronização do código significa definir como variáveis, métodos e classes são definidos, como comentários devem ser organizados,

organização de diretórios etc. O AnEnDB é todo padronizado utilizando as definições encontradas no Guia de Estilo Para Código Python (Python.org). A propriedade coletiva do código foi um dos princípios adotados na definição do projeto junto com os orientadores: alterações no código não são única responsabilidade do único, até o momento, desenvolvedor do *software* e sim responsabilidade de qualquer especialista que precise ou queira participar do projeto.

Banco de dados relacional

As principais tabelas de dados do AnEnDB são: *proteins*, *organisms*, *ecs* (números EC) e *clusters*. Os dados e relacionamentos entre essas quatro tabelas é suficiente para representar os resultados mais importantes: relações de analogia intragenômica e intergenômica por classe de atividade enzimática.

Não há grande complexidade nos relacionamentos entre essas tabelas e o relacionamento mais elaborado é apenas o que existe entre *proteins* e *ecs* (enzimas podem ter uma ou mais classes de atividades enzimáticas e uma classe de atividade enzimática pode ter uma ou mais proteínas relacionadas). Outras tabelas são importantes, pois adicionam informações às analogias encontradas. Por fim, por conta do grande volume de dados, foram criadas tabelas com informações pré-processadas para acelerar o resultado de algumas buscas.

Dadas essas características e a necessidade de simplificar o acesso aos dados a partir de uma única origem (por questões de velocidade de processamento, segurança e maior facilidade para administrar), um banco de dados relacional é utilizado para representar e armazenar todas as estruturas de dados do AnEnDB. A lista completa de tabelas pode ser vista abaixo (Tabela 1):

Tabela 1. Lista de tabelas do banco de dados relacional que compõe o AnEnDB

Nome das tabelas
clustering_methods
clusters
domains
ec_intragenomic_analogy
ec_maps
ec_rectangles
ecs
genome_comparison_clusters
genome_comparisons
kingdoms
map_arrow_coordinates
map_arrows
map_line_coordinates
map_lines
map_polygon_coordinates
map_polygons
map_rectangle_coordinates
map_rectangles
metabolic_pathways
metabolic_pathways_maps
organism_ecs
organism_maps
organisms
pathway_subsystems
pathway_systems
protein_ecs
protein_maps
protein_pdbs
Proteins
similarities
similarity_methods
source_databases
taxonomic_groups_level3
taxonomic_groups_level4

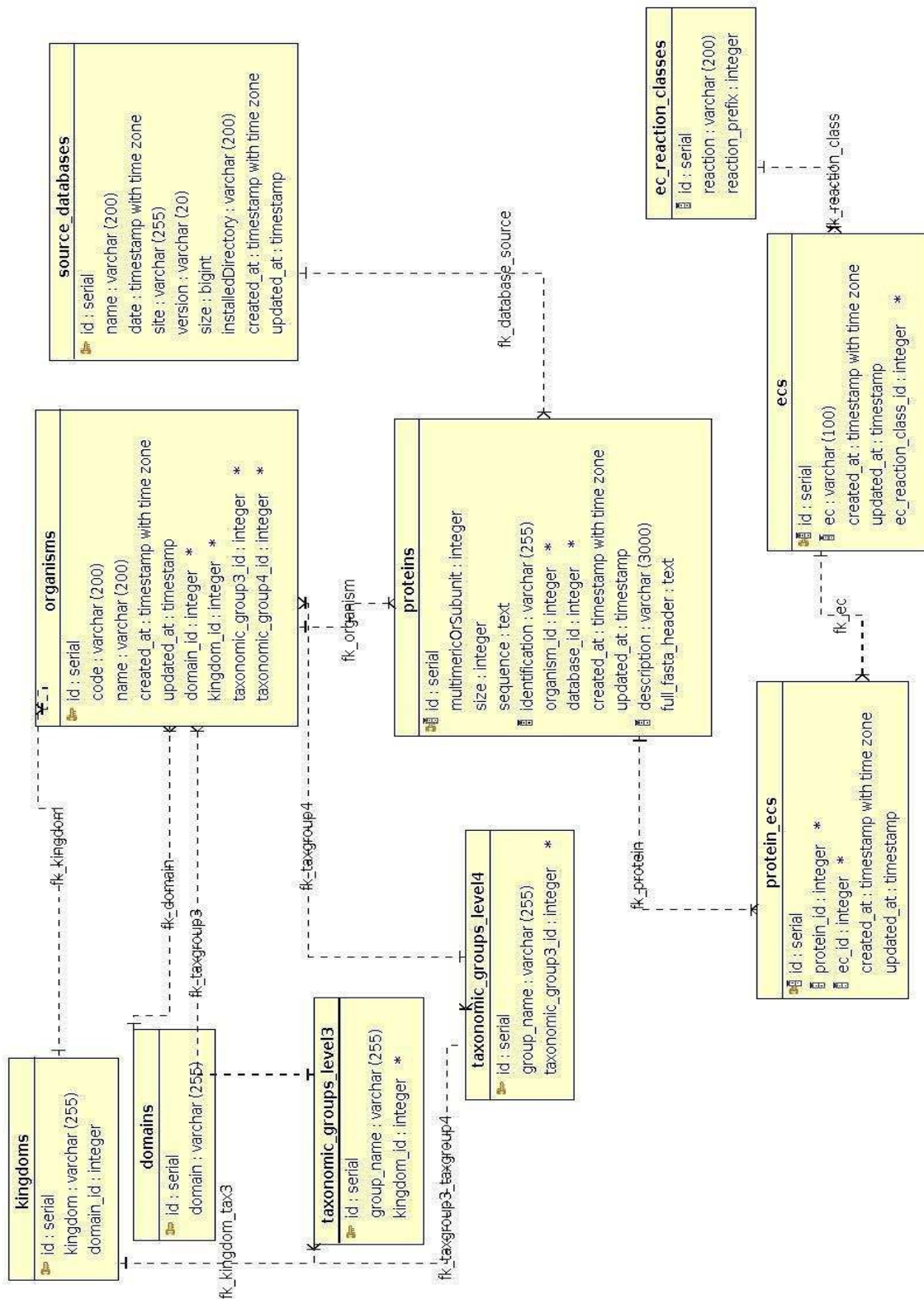


Figura 18. Diagrama que representa os relacionamentos entre as principais tabelas do AnEnDB. Essas tabelas representam os organismos, proteínas, números EC e seus relacionamentos.

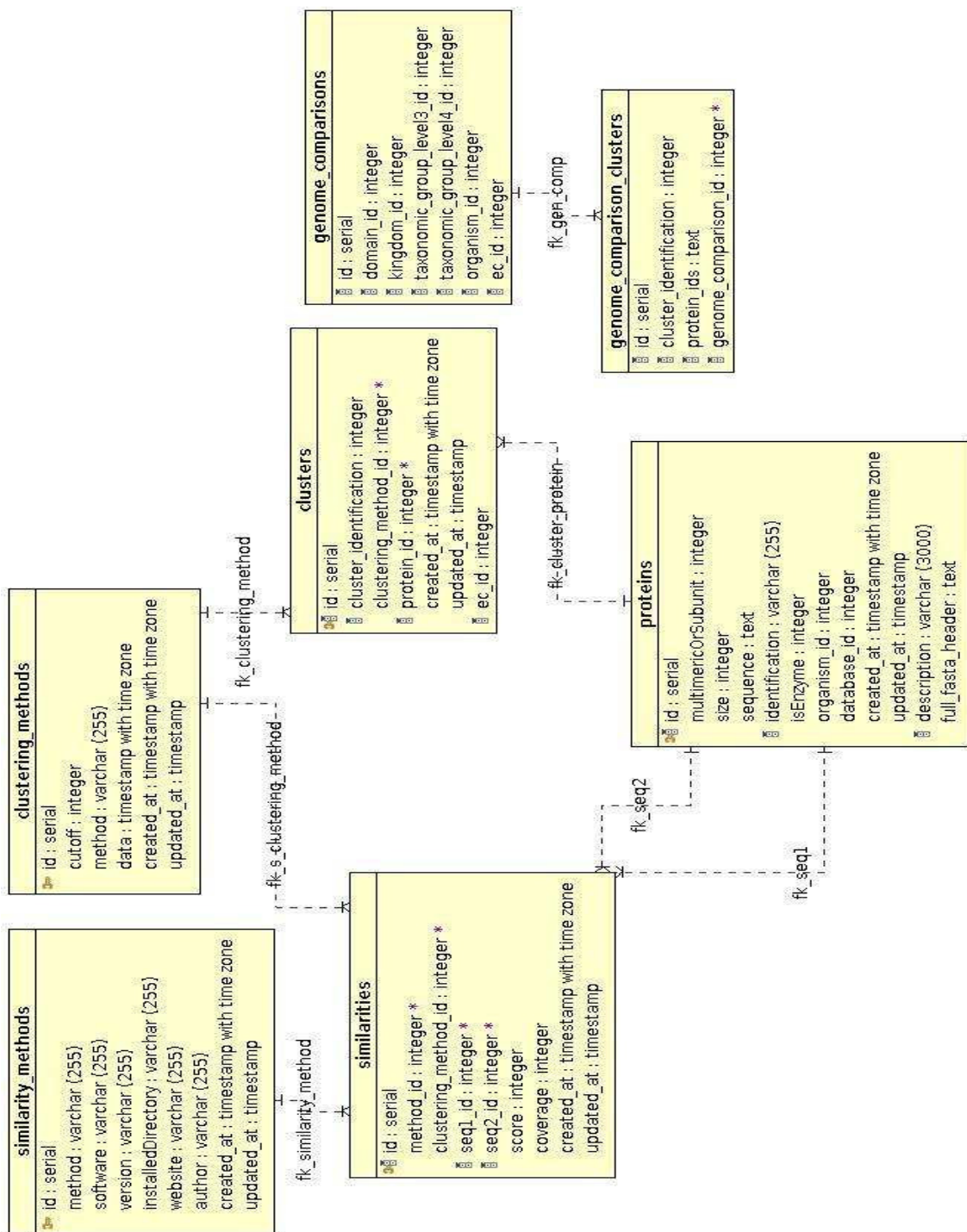


Figura 19. Diagrama que representa as tabelas que registram os grupos de análogos (clusters) e os valores obtidos na busca por similaridade de sequência (similarities). São registrados tanto os grupos de análogos quanto os parâmetros de busca por similaridade e respectivos softwares utilizados para essa busca (similarity_methods e clustering_methods).

AnEnDB: codificação

O código do AnEnDB é dividido em três principais camadas, ilustrados na Figura 20: i) modelagem e generalização da origem primária de dados; ii) processamento de dados e informações no contexto de enzimas análogas; iii) apresentação dos dados através de dois meios distintos.

Todas as três camadas são distribuídas em diversos subcomponentes dentro de suas respectivas categorias.

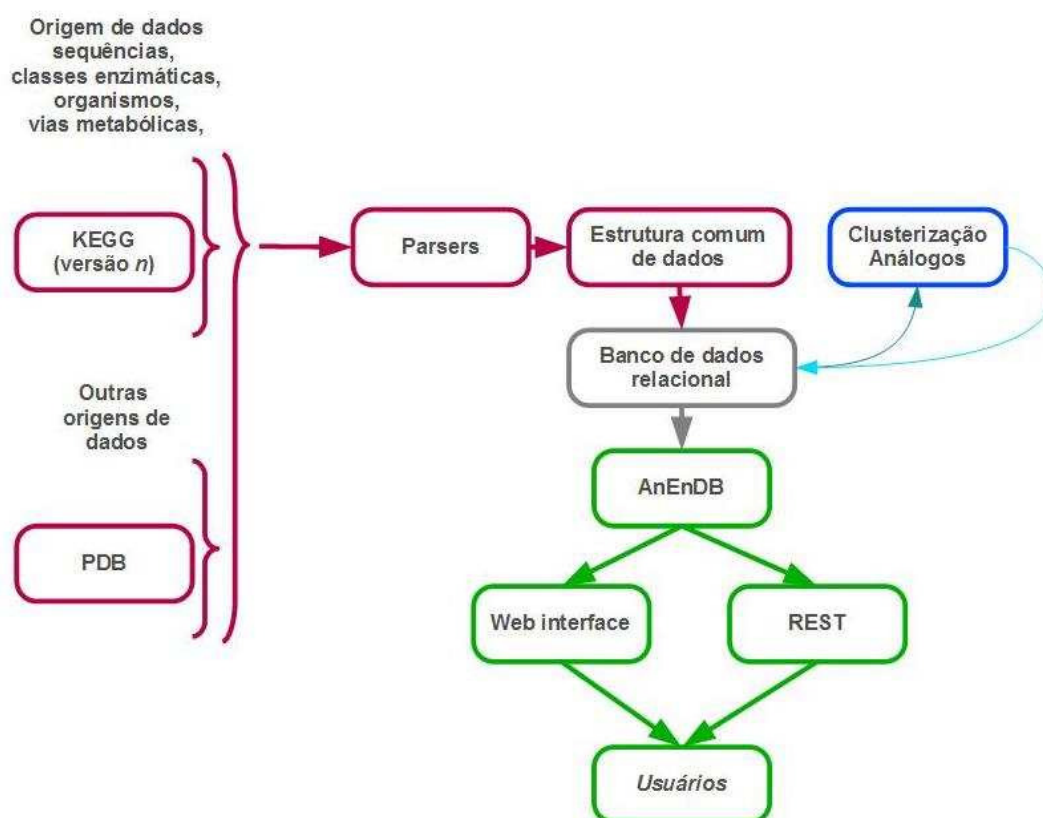


Figura 20. Camadas do AnEnDB. Modelagem e generalização da origem primária de dados (vermelho), processamento de dados e informações no contexto de enzimas análogas (azul) e apresentação dos dados através de dois meios distintos (verde).

Modelagem e generalização da origem primária de dados

A primeira camada do AnEnDB possui três responsabilidades: extrair os dados dos arquivos em formato texto, compor todos os dados em estruturas generalizadas e inserir os dados no banco de dados relacional.

O AnEnDB utiliza em sua primeira versão a origem de dados primária KEGG, versão Fevereiro de 2015. O KEGG possui mais de 200GB de dados distribuídos em arquivos no formato texto ASCII (arquivos Fasta, XML etc) e imagens no formato PNG. Em torno de 6GB de dados são referentes às sequências proteicas e seus identificadores complementados ainda com milhares de arquivos de informações auxiliares distribuídos em diversos formatos texto ASCII (Figura 21 e Figura 22).

A correlação das informações contidas nesses arquivos é processada por classes e métodos executores de processadores de texto: FastaParser, PathwayParser, OrganismParser, PdbParser e EntityParser. A classe FastaParser por exemplo é responsável por apresentar para outros componentes os dados em estruturas nomeadas *proteins* (Figura 23) contendo todos os dados relacionados a proteínas (identificador, sequência, classe de atividade enzimática - se houver -, organismo relacionado e descrição). A camada de processadores de texto e generalização de estrutura de dados servem, portanto como uma interface para os demais componentes do AnEnDB.

```
>mma:MM_2626 DNA-directed RNA polymerase subunit P (EC:2.7.7.6)
MGYKCTRCKQKVEIDYEYTGIRCPYCGHRILVKERPTTIKRIKAE
```

Figura 21. Exemplo de dado para a sequência de aminoácidos da proteína mma:MM_2626 do arquivo T00082.pep do KEGG.

```

mma:MM_0014 ec:2.5.1.31
mma:MM_0028 ec:3.6.4.12
mma:MM_0037 ec:6.3.4.5
mma:MM_0038 ec:6.3.5.5
mma:MM_0039 ec:6.3.5.5
mma:MM_0040 ec:3.6.3.32
mma:MM_2626 ec:2.7.7.6
mma:MM_0047 ec:2.6.1.19
mma:MM_0048 ec:1.2.1.3
mma:MM_0050 ec:6.3.5.7
mma:MM_0056 ec:1.8.98.1
mma:MM_0059 ec:1.2.99.5
mma:MM_0060 ec:1.2.99.5
mma:MM_0070 ec:2.1.1.226
mma:MM_0070 ec:2.1.1.227
mma:MM_0073 ec:5.3.1.6
mma:MM_0074 ec:6.1.1.12
mma:MM_0084 ec:2.7.1.71
mma:MM_0086 ec:2.1.1.206
mma:MM_0087 ec:2.4.2.4

```

Figura 22. Parte do conteúdo do arquivo, do KEGG, `mma_enzyme.list`. Na primeira coluna estão os identificadores de proteínas e na segunda coluna as atividades enzimáticas relacionadas. Em destaque (cor vermelha) está a proteína `mma:MM_2626` demonstrada na Figura 21.

Estrutura *protein* da proteína `mma:MM_2626` modelada pelo AnEnDB e entregue para as outras camadas:

```

{
  'description': 'DNA-directed RNA polymerase subunit P ',
  'ecs': [2.7.7.6],
  'full_fasta_header': '>mma:MM_2626 DNA-directed RNA polymerase \
                        subunit P (EC:2.7.7.6)',
  'identification': 'mma:MM_2626',
  'organism_code': 'mma',
  'organism_name': 'Methanosarcina mazei Gol',
  'sequence': 'MGYKCTRCKQKVEIDYEYTGIRCPYCGHRILVKERPTTIKRIKAE',
  'sequence_size': 45
}

```

Figura 23. Estrutura de dados *protein*, gerada e utilizada pelo AnEnDB. Seus dados (*organism_code*, *identification* etc) são extraídos pelos processadores de texto a partir dos arquivos texto do KEGG.

Todas as outras camadas do AnEnDB lidam apenas com estruturas organizadas como acima, independente de qual banco de dados primário foi utilizado. Portanto, a criação dessa camada permite que a utilização de outro banco de dados primário exija apenas o conhecimento sobre como codificar novos processadores de texto sem afetar qualquer outro processo do AnEnDB.

A última etapa dessa camada é a inserção dos dados no banco de dados relacional. Um conjunto de *scripts* se conecta ao banco e consome as estruturas de dados em direção ao banco de dados relacional. *Scripts* como *slurp_proteins.py*, *slurp_pathways.py*, *slurp_organisms.py* e *slurp_ecs.py* fazem também validações para identificar inconsistências (duplicações de dados, ausência de dados e relacionamentos incorretos) e principalmente definem se os dados serão inseridos através de estruturas orientadas a objeto ou simplesmente gerarão arquivos *script* no formato texto ASCII com instruções SQL para serem executadas manualmente pelo desenvolvedor. A utilização de um ORM impõe uma perda de velocidade significativa dependendo do volume de dados a ser inserido (os dados precisam ser carregados em objetos antes de serem inseridos). Para o KEGG 2015 a inserção de todos os registros de proteínas levaria aproximadamente 15 dias utilizando ORM, enquanto que a geração de arquivos com instruções SQL e posterior inserção manual levou 2 dias. O *script slurp_proteins.py*, portanto, gera arquivos SQL para todas as ~14 milhões de proteínas e exige intervenção manual do desenvolvedor. Já o *script slurp_organisms.py* utiliza estruturas orientadas a objetos e insere diretamente os dados no banco de dados relacional.

Processamento de dados e informações no contexto de enzimas análogas

A segunda principal camada do AnEnDB é responsável por processar as relações de analogia já utilizando os dados contidos no banco de dados relacional.

A principal classe dessa camada é a *OttoCluster* e o principal *script* é o *execute-clustering.py*. A classe *OttoCluster* é a responsável por obter todas as proteínas, agrupá-las em classes de atividades enzimáticas e executar a busca por similaridade entre sequências através do algoritmo de alinhamento local executado pelo BLAST. Além dessas etapas a classe executa o algoritmo de agrupamento de

proteínas de acordo com as premissas estabelecidas por Galperin e colaboradores (1998) (27). O script *execute-clustering.py* obtém as estruturas de dados geradas pela classe *OttoCluster* e adiciona no banco de dados relacional.

A soma das camadas (i) e (ii) do AnEnDB finaliza todo o conjunto de dados relacionados a proteínas, suas atividades enzimáticas, suas analogias e relações taxonômicas em uma macro estrutura de dados representada pelo banco de dados relacional. Portanto, o banco de dados relacional passa a ser a única estrutura de dados organizada e visível para os usuários do AnEnDB.

Apresentação dos dados através de dois meios distintos.

O AnEnDB fornece acesso aos dados do banco de dados relacional de duas formas distintas: interface web, através de formulários de pesquisa e REpresentational State Transfer (REST), através de URLs diretas que retornam resultados de pesquisa.

Interface web

A interface web pode ser acessada utilizando qualquer navegador de internet através do endereço:

`http://157.86.220.224:5000`

Além dos recursos de busca a interface *web* apresenta outras informações relacionadas ao projeto: documentação, contato, referências, e informações gerais sobre o projeto.

Seus principais recursos são as buscas por analogias intragenômicas e intergenômicas orientadas por organismos. Os resultados são mostrados em forma de tabelas ordenáveis e contendo filtros para cada campo de resultado. Por

exemplo, a tabela de resultados de sequências de análogos possui filtros para nomes de organismo e quantidade de sequências nos *clusters*.

Outras características da interface *web* são: filtros e ordenação das tabelas; os campos de busca sugerem resultados (basta digitar a inicial de um organismo para se obter uma lista dos nomes correspondentes e o mesmo para classes de atividade enzimática - números EC); as tabelas de resultado de sequências destacam com cores as sequências de um mesmo organismo; e campos pré-processados como, por exemplo, marcadores de presença ou ausência de analogia intragenômica que podem servir como filtro em tabelas de resultados.

Os resultados seguem sempre o mesmo design gráfico e a navegação termina, caso o usuário deseje, sempre na mesma tabela final que apresenta todos os dados disponíveis sobre uma determinada sequência.

REST: URLs diretas que retornam resultados de pesquisa

O usuário pode obter resultados sem precisar utilizar a interface *web* acessando resultados diretamente por URLs especiais. A utilização de REST garante que o usuário não precise utilizar um navegador ou interagir com a interface, e, além disso, possa fazer seus próprios *scripts*, em qualquer linguagem, acessando resultados através de chamadas diretas ao servidor HTTP. Por exemplo, para acessar todos os dados da proteína *mma:MM_2626* (Figura 24) o usuário pode acessar diretamente a URL:

`http://157.86.220.224:5000/proteins/mma:MM_2626`

```

{
  "proteins": [
    {
      "description": "DNA-directed RNA polymerase subunit P ",
      "domain": "Prokaryotes",
      "ecs": [
        "2.7.7.6"
      ],
      "fasta_header": ">mma:MM_2626 DNA-directed RNA polymerase subunit P (EC:2.7.7.6)",
      "full_fasta_header": ">mma:MM_2626 DNA-directed RNA polymerase subunit P (EC:2.7.7.6)",
      "identification": "mma:MM_2626",
      "kingdom": "Archaea",
      "organism": "Methanosarcina mazei Gol",
      "organism_code": "mma",
      "organism_name": "Methanosarcina mazei Gol",
      "pdb": [],
      "sequence": "MGYKCTRCKQKVEIDYEYTGIRCPYCGHRILVKERPPTTIKRIKAE",
      "sequence_size": 45,
      "taxonomic_group3": "Euryarchaeota",
      "taxonomic_group4": "Methanosarcina"
    }
  ]
}

```

Figura 24. Exemplo de utilização da interface REST. No exemplo, a estrutura de dados retornada provém do acesso à URL:

http://157.86.220.224:5000/proteins/mma:MM_2626

A implementação de novos e mais complexos métodos pode ser facilmente executada pois os principais métodos de análise que fornecem os dados para resposta via REST já estão implementados para a interface via navegação web.

AnEnDB: exploração dos dados

Nesta seção, iremos explorar as informações disponíveis no sistema AnEnDB, apresentando, inicialmente, uma análise descritiva dos dados e, posteriormente, os resultados obtidos em análises realizadas com a finalidade de extrair informações para a construção de um panorama dos processos de convergência ocorridos em atividades enzimáticas ao longo da evolução de espécies representantes dos três domínios da vida (Eukarya, Bacteria, Archaea), bem como demonstrar, através de dois estudos de casos, i) de que forma podemos identificar potenciais novos alvos terapêuticos para o tratamento de doenças infecciosas, através da comparação de atividades enzimáticas compartilhadas entre parasitas e hospedeiros, revelando em quais destas atividades tais organismos utilizam formas distintas de enzimas análogas (analogia intergenômica), e ii) como identificar atividades enzimáticas para as quais distintas formas análogas são codificadas em um mesmo genoma, possibilitando a investigação do envolvimento destas enzimas

em distintos papéis biológicos no organismo que as expressa (analogia intragenômica).

A exploração dos dados utilizará, em alguns resultados, funções disponíveis na classe `Anendb` para, ao mesmo tempo em que expõe resultados, exemplifique a utilização do AnEnDB como ferramenta para gerar scripts personalizados.

O primeiro passo para utilizar a classe `Anendb` é instanciá-la em uma variável (Figura 25).

```
anendb = Anendb ()
```

Figura 25. Exemplo de código que instancia, na variável `anendb`, a classe `Anendb`.

O AnEnDB possui, na versão deste trabalho, 14.258.659 de proteínas (Figura 26) armazenadas em seu banco de dados relacional. Nessa primeira versão o AnEnDB utiliza apenas os dados disponíveis no banco de dados KEGG, versão de fevereiro de 2015. O KEGG, a cada nova versão, aumenta a quantidade de dados disponíveis e atualiza grande quantidade de informações. Por conta dessa característica o AnEnDB não consulta dados e informações do KEGG através de nenhum recurso em tempo real (interface web ou API REST do KEGG). Caso o fizesse o AnEnDB poderia expor informações contraditórias em relação aos dados que estão armazenados estaticamente no banco de dados relacional. Como já foi exposto anteriormente, o AnEnDB possui uma camada especializada em processamento de texto e portanto é possível adicionar futuras versões do KEGG ou outros bancos de dados sendo necessário apenas atualizar ou criar processadores específicos para cada um desses banco de dados.

```
>>> print( anendb.getTotalProteins() )  
14258659
```

Figura 26. Exemplo de código, utilizando o objeto `anendb`, criado no exemplo da Figura 25, que executa o método `getTotalProteins` para retornar o total de proteínas registradas no banco de dados relacional.

Dessas aproximadamente 14 milhões de proteínas, 1.166.610 (Figura 27) possuem atividade enzimática anotada em ao menos uma das 3.829 (Figura 27) classes de atividade enzimática.

```
>>> print( anendb.getTotalEnzymes() )  
1166610  
  
>>> print( anendb.getTotalEcs() )  
3829
```

Figura 27. Exemplo de utilização do código do AnEnDB para obter o total de enzimas e o total de classes de atividade enzimática registradas no banco de dados relacional.

O AnEnDB também possui 3.366 organismos armazenados juntamente com suas taxonomias (de acordo com a classificação taxonômica do KEGG).

```
>>> print( anendb.getTotalOrganisms() )  
3366
```

Figura 28. Exemplo de utilização do código do AnEnDB para obter o total de organismos registrados no banco de dados relacional.

Devido ao contínuo avanço nas tecnologias de sequenciamento e consequente aumento do número de genomas disponíveis, a versão do KEGG utilizada pelo AnEnDB possui uma considerável diferença em termos de quantidade de dados em relação a versões anteriores e que não passam de 10 anos de surgimento (Figura 29). O KEGG em Fevereiro de 2006 possuía 1.227.612 proteínas armazenadas (AnEnDB = 14.258.659), 331 organismos (AnEnDB = 3.366), 224.707 proteínas com classe de atividade enzimática anotada (AnEnDB = 1.166.610) e 2.314 classes de atividades enzimáticas descritas (AnEnDB = 3.829).

Essas características implicam em uma preocupação contínua com relação à otimização e atualização das tecnologias e algoritmos para lidar com o aumento na quantidade de dados. Se os dados do KEGG crescerem em volume na mesma ordem que cresceram nos últimos 10 anos (para registros de proteínas algo em torno de 1.000%) pode ser necessário que o AnEnDB tenha de passar por uma

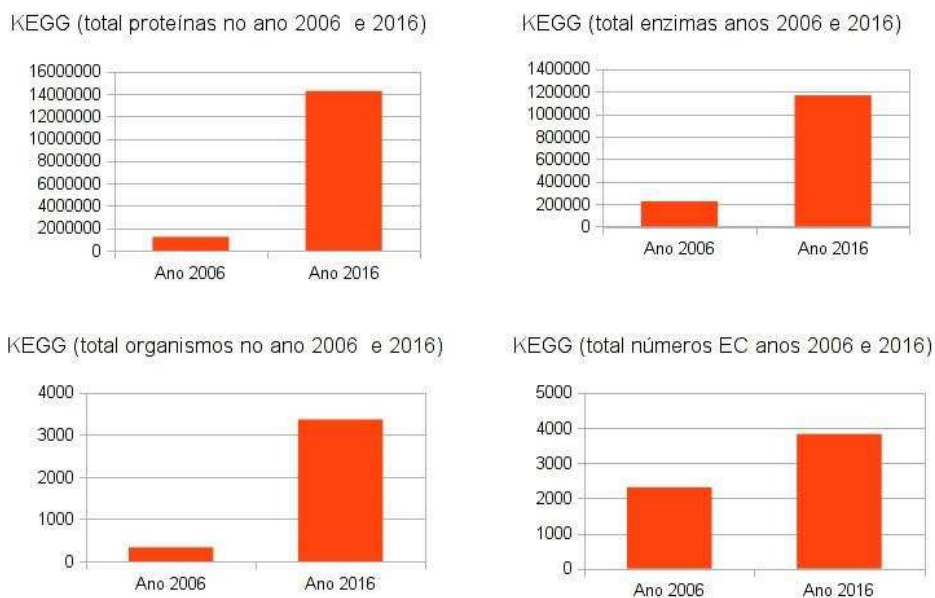


Figura 29: Diferença no volume de dados do KEGG entre os anos de 2006 e 2016. Em sentido horário: total de proteínas armazenadas, total de proteínas com atividade enzimática identificada, total de organismos sequenciados e total de números EC.

reformulação de algumas de suas partes. Isso reforça a importância da definição clara, desde o princípio deste projeto, de uma metodologia de desenvolvimento de *software* e de seu modelo em camadas. Igualmente a vida útil do AnEnDB está diretamente ligada à sua manutenção e atualização contínua.

O AnEnDB utiliza a classificação taxonômica apresentada pelo KEGG. Tal classificação é a mesma classificação hierárquica definida pelo NCBI (73).

O AnEnDB possui métodos responsáveis por retornar informações taxonômicas sobre organismos.

```
>>> print( anendb.getOrganismByName( 'Homo' ) )
[{'code': u'hsa',
  'domain': u'Eukaryotes',
  'kingdom': u'Animals',
  'name': u'Homo sapiens (human)',
  'tax_group3': u'Vertebrates',
  'tax_group4': u'Mammals'}]
```

Figura 30. Exemplo da utilização do método *getOrganismByName* que retorna informações sobre a taxonomia de organismos.

O algoritmo de agrupamento de análogos gerou 17.618 grupos de homólogos, considerando todas as classes de atividade enzimática. Esse número não descreve especificamente organismos ou classes de atividades enzimáticas, mas indica o poder computacional que foi necessário para separar das ~14 milhões de proteínas ~1 milhão de enzimas e realizar uma comparação “todos contra todos” de sequências utilizando o BLAST para então gerarmos 17.618 grupos de enzimas (Figura 31).

```
>>> print( anendb.getTotalGroupsOfHomologous() )
17618
```

Figura 31. Exemplo de utilização do método *getTotalGroupsOfHomologous* que retorna o total de grupos (clusters) gerados pelo AnEnDB.

As cinco classes de atividades enzimáticas que possuem maior quantidade de grupos podem ser vistas na Tabela 2 e Figura 32.

```
>>> anendb.getEcHomologousGroupsStats( order_by_homologous_amount='yes', biggest=5 )
[{'ec': u'2.1.1.-', 'total_clusters': 189, 'total_proteins': 10740},
 {'ec': u'2.3.1.-', 'total_clusters': 198, 'total_proteins': 6279},
 {'ec': u'3.-.-.-', 'total_clusters': 201, 'total_proteins': 1636},
 {'ec': u'1.-.-.-', 'total_clusters': 212, 'total_proteins': 5968},
 {'ec': u'3.1.-.-', 'total_clusters': 241, 'total_proteins': 4353}]
```

Figura 32. Exemplo de código do AnEnDB que retorna as classes de atividades enzimáticas que possuem mais grupos (clusters). O método *getEcHomologousGroupsStats* é configurável através dos parâmetros *order_by_homologous_amount*, que diz se o retorno de

dados é ordenado da maior para a menor quantidade de grupos, e *biggest*, que configura um limite de retorno dos dados.

Tabela 2: Lista das cinco atividades enzimáticas com maior quantidade de clusters registradas no banco de dados relacional.

EC	Total grupos	Total proteínas no EC
3.1.-.-	241	4353
1.-.-.-	212	5968
3.-.-.-	201	1636
2.3.1.-	198	6279
2.1.1.-	189	10740

Classes que possuem '-' em seu código representam atividades enzimáticas cujos tipos/mecanismos de reação química e/ou substratos e/ou cofatores ainda não estão bem definidos. O fato de estas atividades apresentarem um número excessivo de grupos de enzimas indica o esforço que ainda necessita ser feito para determinar com precisão a atividade catalítica de inúmeras enzimas identificadas até o momento. As atividades enzimáticas 1.-.-.-, 2.-.-.-, 3.-.-.-, 4.-.-.-, 5.-.-.- e 6.-.-.-, que reúnem enzimas para as quais somente a classe química de suas reações são conhecidas (oxido-redutases, transferases, hidrolases, liases, isomerases e ligases, respectivamente), estão todas entre as 120 (de 3.829) classes enzimáticas com maior quantidade de grupos formados.

Obter a distribuição dos números EC e enzimas que não possuem todos os níveis de atividade enzimática definidos pode ser útil para garantir precisão no controle dos resultados sobre analogia enzimática. Na Figura 33, é possível observar como essa distribuição ocorre.



Figura 33. Distribuição entre números EC com todos os níveis de classificação enzimática definidos (completos) ou incompletos. O gráfico da esquerda representa a distribuição de **números EC** por classe enzimática, enquanto o gráfico da direita representa a distribuição do número de **enzimas** por classe enzimática.

É importante também identificar as características de cada classe de atividade enzimática. A classe de atividade enzimática que possui maior quantidade de proteínas é a 2.7.7.7 (Figura 34).

```
>>> anendb.getEcHomologousGroupsStats( order_by_proteins_amount='yes', biggest=4 )
{'ec': u'3.6.1.-', 'total_clusters': 140, 'total_proteins': 8305},
{'ec': u'2.1.1.-', 'total_clusters': 189, 'total_proteins': 10740},
{'ec': u'3.6.3.14', 'total_clusters': 105, 'total_proteins': 12949},
{'ec': u'2.7.7.7', 'total_clusters': 83, 'total_proteins': 14146}
```

Figura 34. Exemplo de código do AnEnDB que pode ser utilizado para informar o total de enzimas de uma classe de atividade enzimática. O campo *'total_proteins'* informa o total de proteínas da classe de atividade enzimática referenciada no campo *'ec'*.

O EC 2.7.7.7 é definido pelo *ExPASy* como: *DNA-directed DNA polymerase, Catalyzes DNA-template-directed extension of the 3'-end of a DNA strand by one nucleotide at a time.* Este mesmo número EC é o 24^o (de um total de 3.362) com maior quantidade de grupos de enzimas (possui 83 grupos no total). Ao analisar quantos organismos possuem proteínas anotadas com o EC 2.7.7.7 obtém-se 3.362 de um total de 3.366. Em outras palavras, praticamente todos os organismos registrados no KEGG possuem a classe de atividade enzimática 2.7.7.7. Dada a relevância do tipo de atividade enzimática em 2.7.7.7 (envolvida diretamente na síntese de DNA) não é surpresa que praticamente todos os organismos a possuam.

Com o AnEnDB é possível, por exemplo, identificar, de uma forma simples e direta (via interface web), sem a necessidade de gerar *scripts* próprios, se um determinado organismo possui analogia intragenômica (Figura 35), ou seja, formas distintas de enzimas com a mesma atividade enzimática coexistindo em um mesmo organismo.

```
>>> anendb.getOrganismIntragenomicAnalogy( organism_name='Homo' )
{'hsa': ['1.1.1.14',
        '1.1.1.239',
        '1.1.1.35',
        '1.1.1.64',
        '1.10.2.2',
        '1.11.1.6',
        ...]}
```

Figura 35. Exemplo, resumido, da informação retornada pelo método *getOrganismIntragenomicAnalogy* do AnEnDB, que retorna as classes de atividade enzimática para um determinado organismo (Homo sapiens, no exemplo) que possuem analogia intragenômica.

Da mesma maneira é possível identificar quais organismos no total possuem analogia intragenômica (Figura 36). O AnEnDB identificou 2.247 organismos com analogia intragenômica em pelo menos uma atividade enzimática:

```
>>> totOrgsWithIntragenomiAnalogy = 0
>>>
>>> organisms = anendb.getOrganismsWithIntragenomicAnalogy()
>>> print( len( organisms ) )
2247
```

Figura 36. Exemplo de código do AnEnDB que retorna o total de organismos que possuem analogia intragenômica. O método *getOrganismsWithIntragenomicAnalogy* retorna uma lista de organismos (que possuem analogia intragenômica) que é contada pelo método interno Python *len*.

É interessante notar que, dos 3.366 organismos registrados no AnEnDB, em torno de 67% (2.247) possuem analogia intragenômica (Figura 37 e Figura 39).

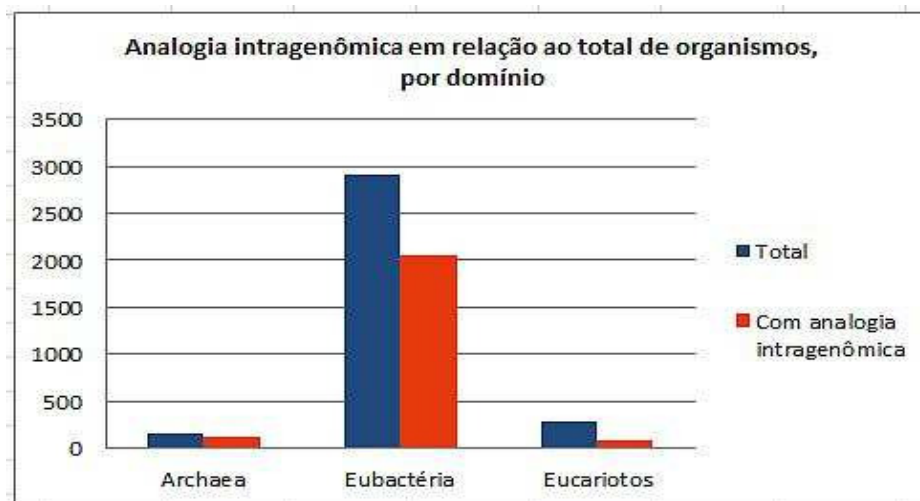


Figura 37. Relação, por domínio, entre organismos que possuem analogia intragenômica em relação ao total de organismos registrados no KEGG.

Esse fato reforça a observação de que analogia não é um fenômeno raro e também a necessidade da contínua busca por compreender melhor a origem e implicações deste fenômeno, particularmente em vias metabólicas. Por outro lado, o fato de inúmeros organismos codificarem em seus genomas formas distintas de enzimas análogas nos faz pensar na possibilidade de tais formas enzimáticas estarem desempenhando distintos papéis biológicos nestes organismos, ao invés de representarem somente uma redundância funcional. E de fato, esse é o tema de outro projeto de pesquisa desenvolvido em nosso grupo. Um dado importante a ser observado é o fato de a classificação taxonômica do KEGG (versão de fevereiro de 2015), agrupar archaea e eubacteria num único domínio "Procariotos".



Figura 39. Percentual de analogia intragenômica, a partir dos organismos registrados no KEGG, nos domínios Eucarioto, Eubactéria e Archaea.

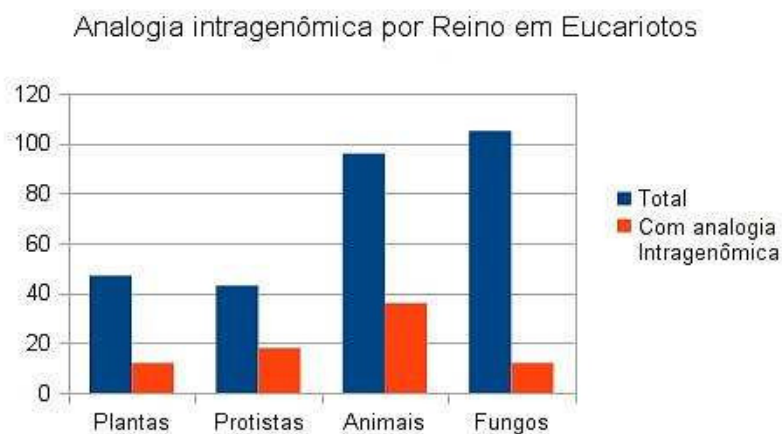


Figura 38. Relação de analogia intragenômica em comparação ao total de organismos registrados no KEGG, agrupados por reinos do domínio Eucarioto.

Existem 1.712 atividades enzimáticas nas quais todas as enzimas foram reunidas em um único grupo, após a “clusterização” pelo pipeline AnEnPi-v2 (Figura 40).


```

>>> clusters = anendb.getClustersByEc()
>>> withSingleCluster = []
>>>
>>> for cluster in clusters:
...     if cluster['total_clusters'] == 1:
...         withSingleCluster.append( cluster )
...
>>>
>>> print( str( len( withSingleCluster ) ) )
1712

```

Figura 40. Exemplo de código que busca em todas as classes de atividades enzimáticas aquelas que possuem apenas um grupo (cluster).

Esse resultado indica que não há analogia intragenômica ou intergenômica detectável nessas atividades enzimáticas. A distribuição dessas classes de atividades enzimáticas em termos de números EC completos e incompletos auxilia, por exemplo, na melhor caracterização de resultados (conclusões feitas sobre enzimas com EC incompleto precisam de validação mais rigorosa). (Figura 41).

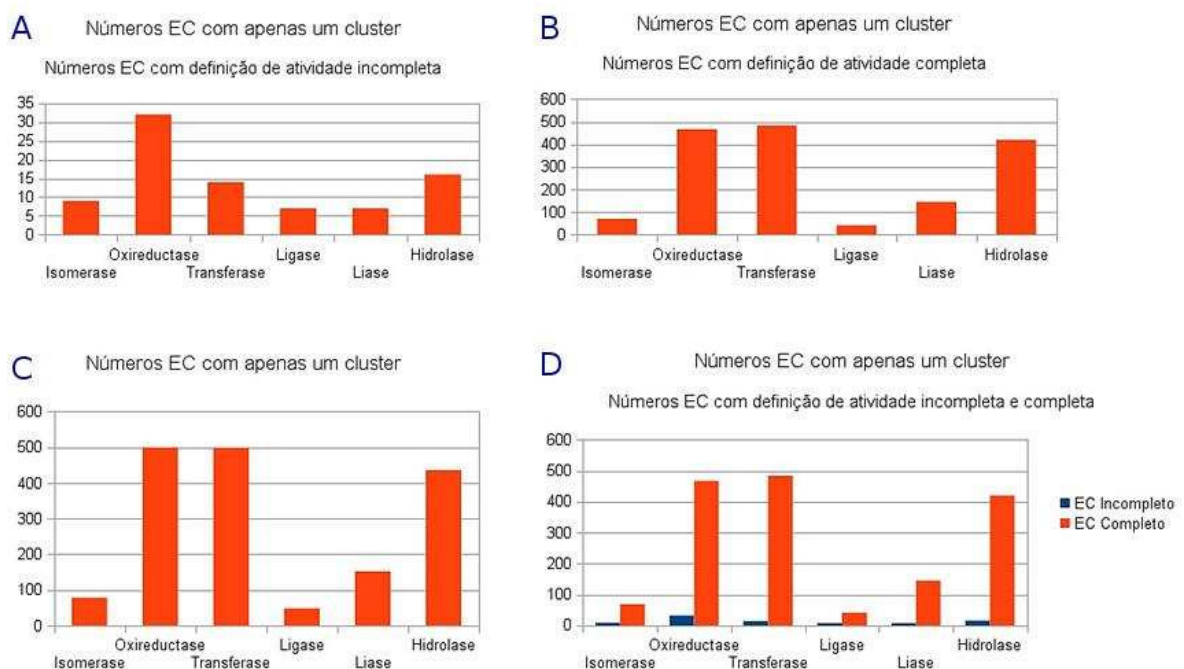


Figura 41. Distribuição entre atividades enzimáticas (representadas por números EC) nas quais apenas um único grupo de enzimas foi formado após o agrupamento pelo *pipeline* AnEnPi-v2. (A) números EC que não possuem todos os níveis de atividade definidos; (B) números EC que possuem todos os níveis definidos; (C) números EC independente de estarem completamente definidos ou não; (D) comparação entre números EC completos e incompletos.

A exploração de dados apresentada anteriormente não busca encontrar respostas específicas, mas sim demonstrar que o AnEnDB pode ser utilizado como uma ferramenta útil para extrair e analisar informação relevante e que possa apontar estudos mais aprofundados sobre evolução e demais aspectos relacionados a enzimas análogas.

É importante ressaltar que como todo *software* o AnEnDB não é um produto acabado nele mesmo. *Softwares* possuem tempo de vida útil e, ao longo desse tempo, novas funções e novas abordagens podem ser implementadas. A principal força do AnEnDB é fornecer um ambiente onde essas novas abordagens possam ser implementadas com o mínimo de esforço possível.

Estudo de caso: analogia intergenômica entre *Trypanosoma cruzi* e *Homo sapiens*

Como pode ser revelado por uma busca no AnEnDB (Figura 42 e Figura 43), o *T. cruzi* pertence à classe Euglenozoa e ordem Kinetoplastida (de acordo com a classificação taxonômica utilizada pelo KEGG).

The screenshot shows a search interface with a search bar containing 'Seach' and an 'Organism:' field with 'Tryp' entered. Below the search bar is a table with the following structure:

Eukaryotes	Protists	Euglenozoa	Kinetoplasts	tcr	Trypanosoma cruzi	Has 14 EC(s) with: 1.10.2.2 1.6.5.5 2.7.7.6 2.7.7.7 3.1.3.11 3.1.3.16 3.1.3.48 3.6.1.1 3.6.3.14 5.2.1.8 5.99.1.2 6.1.1.5 6.3.2.19 6.5.1.1
------------	----------	------------	--------------	-----	-------------------	---

Figura 42. Exemplo de um resultado de da busca feita através da interface web do AnEnDB para organismos cujo nome científico possua a *string* "Tryp".

```

>>> anendb.getOrganismByName( name='cruzi' )
>>>
[{'code': u'tcr',
  'domain': u'Eukaryotes',
  'kingdom': u'Protists',
  'name': u'Trypanosoma cruzi',
  'tax_group3': u'Euglenozoa',
  'tax_group4': u'Kinetoplasts'}]

```

Figura 43. Exemplo de código do AnEnDB que retorna dados dos organismos que possuem no nome científico a string “cruzi”

É bastante útil ao lidar com o AnEnDB guardar os códigos identificadores dos organismos. Códigos de organismos evitam problemas quando é necessário ser específico com vírgulas, espaços, e parênteses, presentes em nomes completos e por extenso dos organismos. O AnEnDB utiliza como parâmetro tanto nomes completos como códigos, porém, recomenda-se utilizar o código ao invés do nome.

Inúmeros tripanosomatídeos são responsáveis por doenças importantes em seres humanos como a doença do sono (*T. brucei*), leishmaniose (*Leishmania sp.*) e doença de Chagas (*T. cruzi*) (74).

A doença de chagas é um importante problema de saúde pública e afeta, até o momento deste trabalho, entre 2 e 3 milhões de pessoas no Brasil, aproximadamente o equivalente (numericamente) a toda a população do Uruguai ou da Jamaica (75), ou das cidades de Belo Horizonte, Recife e Salvador (76). Se forem consideradas todas as Américas a doença possui aproximadamente 12 milhões de portadores (77).

A doença de Chagas foi primeiramente descrita por Carlos Chagas em 1909 e continua sendo estudada continuamente devido ao seu impacto na saúde pública de diversos países. O sequenciamento completo de *T. cruzi* foi finalizado em 2005.

Por se tratar de uma busca por analogia intergenômica entre *T. cruzi* e *H. sapiens*, é importante consultar os dados relativos a *H. sapiens* no AnEnDB (Figura 44 e Figura 45).

```

>>> anendb.getOrganismByName ( name='Homo' )
>>>
[{'code': u'hsa',
  'domain': u'Eukaryotes',
  'kingdom': u'Animals',
  'name': u'Homo sapiens (human)',
  'tax_group3': u'Vertebrates',
  'tax_group4': u'Mammals'}]

```

Figura 44. Exemplo de código do AnEnDB que retorna os dados dos organismos que possuem no nome científico a *string* “Homo”.

Search

Organism:

Organism(s)

Domain	Kingdom	Taxonomic Group 3	Taxonomic Group 4	Code	Name	Intragenomic Analogy
Eukaryotes	Animals	Vertebrates	Mammals	hsa	Homo sapiens (human)	Has 137 EC(s) with: 1.1.1.14 1.1.1.239 1.1.1.35 1.1.1.64 1.10.2.2 1.11.1.6 1.11.1.9 1.14.11.27 1.15.1.1 1.16.3.1 1.2.1.3 1.3.1.20 1.3.1.24 1.3.1.38 1.3.5.1 1.5.1.9 1.6.5.3 1.6.99.3 1.8.3.2 1.9.3.1 2.1.1.43 2.1.1.6 2.1.1.67 2.3.1.15

Figura 45. Tela que mostra o resultado de da busca pela através da interface web do AnEnDB para organismos cujo nome científico que possuem possua a *string* Homo.

O AnEnDB, em sua interface web, dispõe de um recurso de busca por analogia intergenômica (Figura 46), ou seja, compara as vias metabólicas e atividades enzimáticas anotadas em um par de organismos selecionados e

apresenta aquelas atividades nas quais formas enzimáticas análogas são codificadas nos genomas dos organismos analisados.

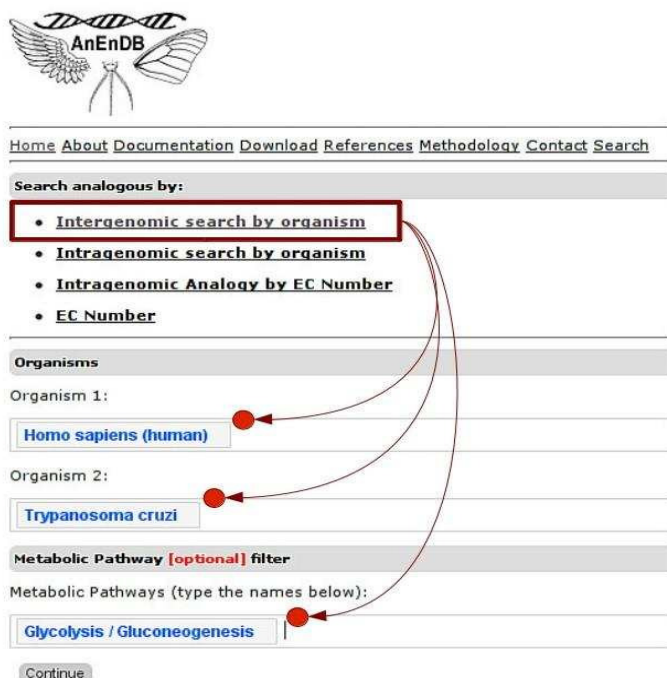


Figura 46. Tela da interface web do AnEnDB mostrando a opção de executar uma busca por analogia intergenômica. A figura representa a busca por analogia intergenômica entre as espécies *H. sapiens* e *T. cruzi*. A interface oferece opcionalmente um filtro por via metabólica.

Caso não seja utilizado o filtro por via metabólica, as atividades enzimáticas (representadas por números EC) comuns ao *T. cruzi* e *H. sapiens* nas quais foi possível detectar analogia intergenômica são as seguintes, conforme as Figura 47 e Figura 48 : 1.6.99.3, 2.7.1.2, 3.6.1.23, 5.3.3.2 e 6.5.1.3.

```
>>> commonEcs = anendb.getCommonEcsOfOrganism( organism_codes="tcr, hsa" )
>>>
>>> print( commonEcs['analogy_only_ec_numbers'] )
['1.6.99.3', '2.7.1.2', '3.6.1.23', '5.3.3.2', '6.5.1.3']
```

Figura 47. Exemplo de código do AnEnDB que retorna as classes de atividade enzimática com analogia intergenômica entre os organismos "tcr" (*T. cruzi*) e "hsa" (*H. sapiens*).

Organisms

Organism 1: Homo sapiens (human)

Organism 2: Trypanosoma cruzi

Check all in all tables

Common EC numbers between 'Organism 1 and Organism 2' that has analogy only.

Oxyreductase	Transferase	Hydrolase	Lyase	Isomerase	Ligase
<input type="checkbox"/> 1.6.99.3	<input type="checkbox"/> 2.7.1.2	<input type="checkbox"/> 3.6.1.23		<input type="checkbox"/> 5.3.3.2	<input type="checkbox"/> 6.5.1.3

Figura 48. Lista de classes de atividades enzimáticas com analogia (exclusivamente) intergenômica entre *H. sapiens* e *T. cruzi*.

Caso seja utilizado o filtro para a via metabólica Glicólise/Gliconeogênese (como na Figura 49) o resultado da busca se reduz a apenas as atividades enzimáticas que possuem analogia intergenômica entre *H. sapiens* e *T. cruzi* e que façam parte da via metabólica selecionada (Figura 49).

Organisms

Organism 1: Homo sapiens (human)

Organism 2: Trypanosoma cruzi

The list of ECs was filtered by the metabolic pathways below:
 [Glycolysis / Gluconeogenesis](#)

Check all in all tables

Common EC numbers between 'Organism 1 and Organism 2' that has analogy only.

Oxyreductase	Transferase	Hydrolase	Lyase	Isomerase	Ligase
	<input type="checkbox"/> 2.7.1.2				

Figura 49. Lista de classes de atividades enzimáticas com analogia (exclusivamente) intergenômica entre *H. sapiens* e *T. cruzi*, utilizando também o filtro pela via metabólica Glicólise/Gliconeogênese. O único número EC retornado é o 2.7.1.2, pois corresponde a única atividade enzimática da via glicolítica que possui analogia intergenômica entre *H. sapiens* e *T. cruzi*.

O resultado da pesquisa, como apresentado na Figura 49 , exibe um *link* (“-*Glycolisis / Glucogenesis* -”) para abrir o mapa da via metabólica Glicólise/Gliconeogênese com sobreposição das atividades enzimáticas anotadas em *H. sapiens* e *T. cruzi*. Dessa maneira, é possível identificar quais são as classes de atividade enzimática exclusivas de cada organismo na via metabólica selecionada (Figura 50).

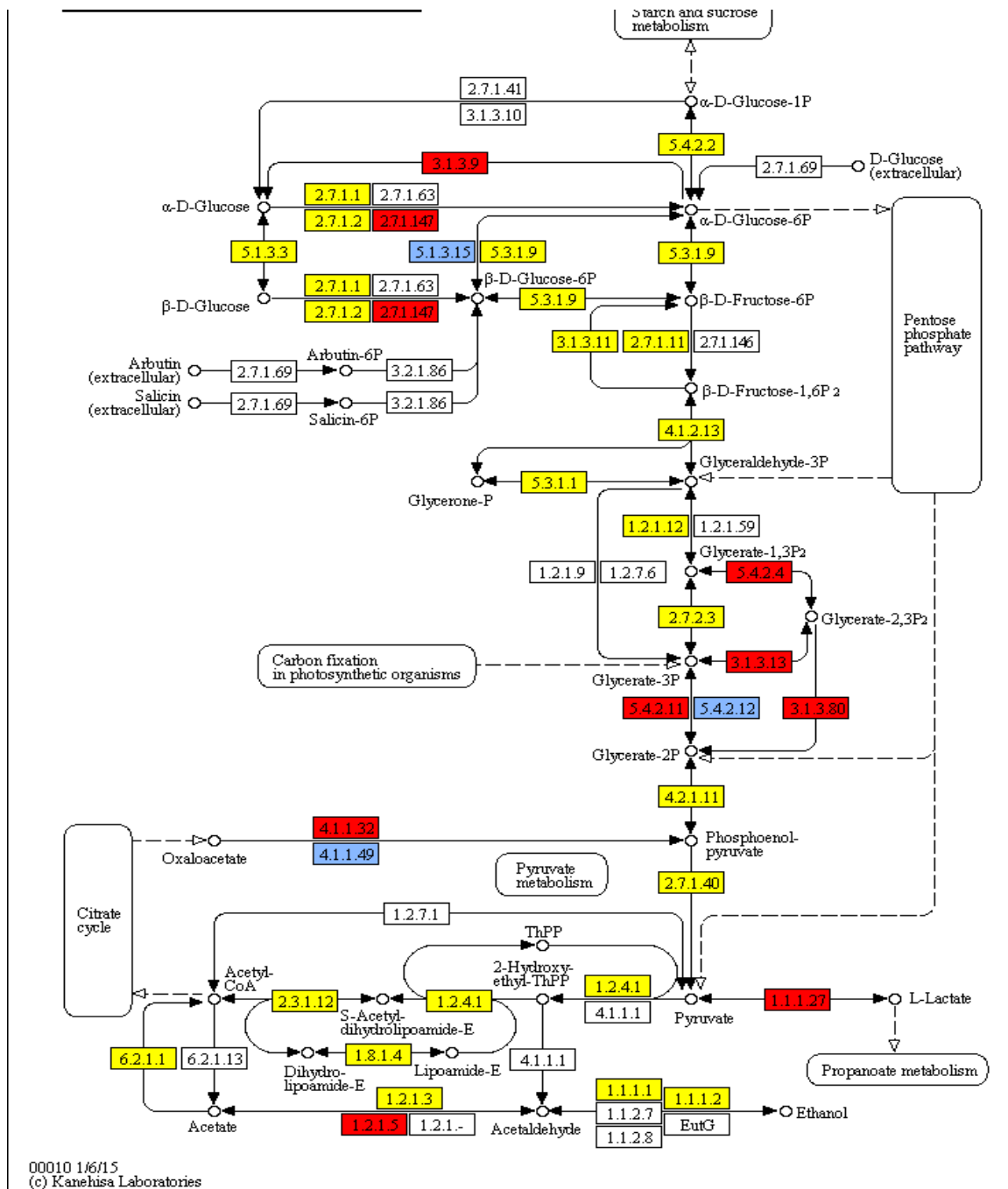


Figura 50. Via metabólica Glicólise/Gliconeogênese representando as atividades enzimáticas anotadas nos genomas dos organismos *H. sapiens* e *T. cruzi*. As atividades enzimáticas em amarelo são as atividades compartilhadas entre os dois organismos. As

Ao selecionar o único número EC (2.7.1.2) na via glicolítica com analogia (exclusivamente) intergenômica para os organismos em questão obtém-se a lista de grupos (*clusters*) aos quais pertencem, respectivamente, a forma enzimática humana e a forma enzimática de *T. cruzi* nesta atividade enzimática (Figura 51).

EC number: 2.7.1.2	
Organism	clusters
Homo sapiens (human) Eukaryotes Animals Vertebrates Mammals	cluster id 4143 (1)
Trypanosoma cruzi Eukaryotes Protists Euglenozoa Kinetoplasts	cluster id 4142 (1)

Figura 51. Grupos (clusters) aos quais pertencem as formas análogas entre *H. sapiens* e *T. cruzi* na atividade enzimática 2.7.1.2 da via glicolítica.

Ambos os grupos listados na Figura 51 possuem apenas uma sequência (enzima) conforme o número indicado entre parênteses do resultado. Ao clicar no *cluster* é possível exibir as informações sobre as sequências (Figura 52).

Protein Identification	Fasta Header	Organism	Sequence	PDB
tcr:510187.100	>tcr:510187.100 glucokinase 1 (EC:2.7.1.2)	Trypanosoma cruzi Eukaryotes Protists Euglenozoa Kinetoplasts	MNKELSHELCEELKTPAWNVPLTFVGDVGGTSARMGFVREGKNDVHACVTRYSHKRKD ITELIEFFNEIIEMLPASVVKRVKAGVINVPGPVYGGAVGGPFNNLKGARLSDYFKALFP PGRSAILNDLEAGFGVLAVSDAHVFSEYFVGMWEGTQWRTCEQEPAGSVGRGRCLLVLP GTGLGSSLYYNPMNQHQHVPLELGSQTIPHRKDDIDYIQTUHAELKLLPNYENMVSAGL EFHYRQVVRGSRPPCSAGEIAKLASEGDANACAMKKYHEYLMRVGSSEASHLLPLTVLV GDNIVNVAFFYRNPQNLKEMHREALNHEMERFGFQSRVTYLRQKLLNLLNLMGCRYRGLDL SVGKKQKAQL	2Q2R

Figura 52. Única sequência do cluster número 4142 do organismo *T. cruzi* da classe de atividade enzimática 2.7.1.2.

Um importante passo nesse tipo de análise é caracterizar as proteínas envolvidas na comparação entre genomas. A busca de informações sobre a via metabólica selecionada, classificação estrutural das sequências encontradas e relações evolutivas são fundamentais para validar o resultado como sendo relevante para uma pesquisa mais aprofundada. A identificação de domínios, por exemplo, das sequências proteicas, pode ajudar a identificar mais detalhes sobre a proteína ou simplesmente refutá-la como objeto de estudo. O EC 2.7.1.2 (apresentado no resultado da busca) oferece imediatamente, a partir dos dados do KEGG, informação adicional do identificador PDB. Esse tipo de informação é útil para iniciar um estudo de comparação entre estruturas 3D de proteínas e, portanto, é um dos passos utilizados na investigação por possíveis alvos moleculares para o desenvolvimento de novos fármacos. Naturalmente as informações sobre estrutura das proteínas, a partir do identificador PDB, não são suficientes para justificar o achado de um potencial novo alvo terapêutico, porém, é um dado significativo que auxilia em pesquisas caráter dessa natureza.

O propósito deste trabalho é demonstrar que o AnEnDB pode ser útil como fonte de dados e, ao mesmo tempo, assinalar que qualquer estudo sobre analogia implica num aprofundamento em relação aos dados obtidos. Qualquer que seja o estudo proposto utilizando o AnEnDB, este precisa ser contextualizado e é natural que novas funções para o AnEnDB tenham que ser implementadas ou atualizadas.

Estudo de caso: analogia intragenômica em *Homo sapiens*

Analogia intragenômica enzimática ocorre quando um organismo, para uma mesma classe de atividade enzimática (número EC), codifica mais de uma forma enzimática com diferenças significativas em suas estruturas tridimensionais. A observação deste evento levanta algumas questões importantes como, por exemplo: seriam essas atividades enzimáticas um tipo de redundância funcional? Ou são expressas de maneira diferente? Se forem diferencialmente expressas, de que forma isto ocorre quantitativamente (níveis de expressão), espacialmente (órgãos, tecidos, tipos celulares, compartimentos celulares) e temporalmente (estágio de desenvolvimento, ciclo de vida, ciclo circadiano, resposta a estímulos externos)?

Nesse caso, quais seriam seus mecanismos de regulação? O surgimento dessas enzimas análogas e/ou atividades enzimáticas estariam relacionadas a eventos geológicos conhecidos (como por exemplo o “envenenamento” da atmosfera primitiva com oxigênio)? Seriam essas formas enzimáticas provenientes de linhagens ancestrais conhecidas?

Para identificar analogia intragenômica com o AnEnDB um primeiro passo pode ser obter os ECs que possuem analogia intragenômica predita em um determinado organismo (Figura 53).

```
>>> ecs = anendb.getOrganismIntragenomicAnalogy( organism_code='hsa' )
>>> print( len( ecs['hsa'] ) )
137
```

Figura 53. Exemplo de código do AnEnDB que retorna as classes de atividade enzimática que possuem analogia intragenômica em um determinado organismo. No exemplo, o método *getOrganismIntragenomicAnalogy* retorna as classes de atividade enzimática do organismo cujo código é “hsa” (*H. sapiens*) e em seguida apenas retorna o total de classes encontradas (137).

Também é possível obter essa informação diretamente da busca por organismo através da interface web (Figura 54).

Seach

Organism:

Organism(s)

Domain	Kingdom	Taxonomic Group 3	Taxonomic Group 4	Code	Name	Intragenomic Analogy
Eukaryotes	Animals	Vertebrates	Mammals	hsa	Homo sapiens (human)	Has 137 EC(s) with. 1.1.1.14 1.1.1.239 1.1.1.35 1.1.1.64 1.10.2.2 1.11.1.6 1.11.1.9 1.14.11.27 1.15.1.1 1.16.3.1 1.2.1.3 1.3.1.20 1.3.1.24 1.3.1.38 1.3.5.1 1.5.1.9 1.6.5.3 1.6.99.3 1.8.3.2 1.9.3.1 2.1.1.43 2.1.1.6 2.1.1.67 2.3.1.15 2.3.1.181 2.3.1.20 2.3.1.23 2.3.1.4

Figura 54. A busca por organismos apresenta a lista de EC que possuem analogia intragenômica.

No exemplo da Figura 54 foram encontrados 137 ECs com analogia intragenômica em *H. sapiens*. O próximo passo consiste em identificar os grupos de enzimas e obter as sequências. Na Figura 55 é utilizado arbitrariamente o EC 5.3.99.2.

```

>>> clusters = anendb.getClustersByEcAndOrganism( \
          ec='5.3.99.2', organism_code='hsa' )

>>> print( clusters['cluster_identifications'] )
[16257, 16258]

>>> clusters = clusters['cluster_identifications']

>>> proteins = anendb.getClusterProteinsOfOrganisms( \
          cluster_ids=clusters, organism='hsa' )
>>> proteins[16257]['proteins'][0]['identification']
u'hsa:5730'

>>> proteins[16257]['proteins'][0]['sequence']
MATHHTLWMGLALLGVLDLQAAPEAQVSVQPNFQQDKFLGRWFSAGLAS
NSSWLREKKAALSMCKSVVAPATDGGGLNLTSTFLRKNQCETRTMLLQPAG
SLGSYSYRSPHWGSTYSVSVVETDYPDQYALLYSQGSKGPGEDFRMATLYS

```

Figura 55. Exemplo de exploração, através do código do AnEnDB, dos dados de analogia intragenômica para a classe de atividade enzimática 5.3.99.2. O primeiro método (*getClustersByEcAndOrganism*) obtém os grupos (clusters) relativos ao organismo “hsa” (*H. sapiens*) do EC 5.3.99.2. Em seguida retorna os identificadores dos grupos encontrados (no exemplo, os identificadores 16257 e 16258). Mais adiante, retorna as sequências do grupo 16257 (no exemplo existe apenas a primeira sequência).

Através da interface web o processo se inicia abrindo o recurso *Intragenomic search by organism* (Figura 56).

Search analogous by:

- [Intergenomic search by organism](#)
- **[Intragenomic search by organism](#)**
- [Intragenomic Analogy by EC Number](#)
- [EC Number](#)

Organisms

Organism:

Figura 56. Tela do AnEnDB para iniciar busca por analogia intragenômica.

Os resultados são os mesmos obtidos pela codificação manual das funções como pode ser observado na Figura 57.

Organism: Homo sapiens (human)

Check all in all tables

EC numbers with intragenomic analogy.

Oxyreductase	Transferase	Hydrolase	Lyase	Isomerase	Ligase
<input type="checkbox"/> 1.1.1.14	<input type="checkbox"/> 2.1.1.43	<input type="checkbox"/> 3.1.1.13	<input type="checkbox"/> 4.1.1.15	<input type="checkbox"/> 5.1.99.4	<input type="checkbox"/> 6.1.1.21
<input type="checkbox"/> 1.1.1.239	<input type="checkbox"/> 2.1.1.6	<input type="checkbox"/> 3.1.1.29	<input type="checkbox"/> 4.2.1.2	<input type="checkbox"/> 5.2.1.8	<input type="checkbox"/> 6.3.2.19
<input type="checkbox"/> 1.1.1.35	<input type="checkbox"/> 2.1.1.67	<input type="checkbox"/> 3.1.1.3	<input type="checkbox"/> 4.2.99.18	<input type="checkbox"/> 5.3.3.12	<input type="checkbox"/> 6.3.2.2
<input type="checkbox"/> 1.1.1.64	<input type="checkbox"/> 2.3.1.15	<input type="checkbox"/> 3.1.1.31		<input type="checkbox"/> 5.3.99.2	<input type="checkbox"/> 6.4.1.4
<input type="checkbox"/> 1.10.2.2	<input type="checkbox"/> 2.3.1.181	<input type="checkbox"/> 3.1.1.4		<input type="checkbox"/> 5.3.99.3	
<input type="checkbox"/> 1.11.1.6	<input type="checkbox"/> 2.3.1.20	<input type="checkbox"/> 3.1.1.47		<input type="checkbox"/> 5.4.99.12	
<input type="checkbox"/> 1.11.1.9	<input type="checkbox"/> 2.3.1.23	<input type="checkbox"/> 3.1.1.5		<input type="checkbox"/> 5.99.1.2	
<input type="checkbox"/> 1.14.11.27	<input type="checkbox"/> 2.3.1.4	<input type="checkbox"/> 3.1.1.56			
<input type="checkbox"/> 1.15.1.1	<input type="checkbox"/> 2.3.1.48	<input type="checkbox"/> 3.1.11.2			
<input type="checkbox"/> 1.16.3.1	<input type="checkbox"/> 2.3.1.50	<input type="checkbox"/> 3.1.13.4			
<input type="checkbox"/> 1.2.1.3	<input type="checkbox"/> 2.3.1.51	<input type="checkbox"/> 3.1.2.2			
<input type="checkbox"/> 1.3.1.20	<input type="checkbox"/> 2.3.1.76	<input type="checkbox"/> 3.1.26.4			
<input type="checkbox"/> 1.3.1.24	<input type="checkbox"/> 2.3.2.13	<input type="checkbox"/> 3.1.26.5			
<input type="checkbox"/> 1.3.1.38	<input type="checkbox"/> 2.3.2.4	<input type="checkbox"/> 3.1.3.16			
<input type="checkbox"/> 1.3.5.1	<input type="checkbox"/> 2.4.1.149	<input type="checkbox"/> 3.1.3.2			
<input type="checkbox"/> 1.5.1.9	<input type="checkbox"/> 2.4.1.198	<input type="checkbox"/> 3.1.3.3			
<input type="checkbox"/> 1.6.5.3	<input type="checkbox"/> 2.4.1.22	<input type="checkbox"/> 3.1.3.36			
<input type="checkbox"/> 1.6.99.3	<input type="checkbox"/> 2.4.1.255	<input type="checkbox"/> 3.1.3.4			
<input type="checkbox"/> 1.8.3.2	<input type="checkbox"/> 2.4.1.69	<input type="checkbox"/> 3.1.3.48			
<input type="checkbox"/> 1.9.3.1	<input type="checkbox"/> 2.4.1.83	<input type="checkbox"/> 3.1.3.5			
	<input type="checkbox"/> 2.4.2.26	<input type="checkbox"/> 3.1.3.56			

Figura 57. Resultado mostrando a lista de classes de atividade enzimática que possuem analogia intragenômica em H. sapiens, utilizando a interface web do AnEnDB.

Ao selecionar as classes de atividade enzimática e submeter uma busca, a interface lista os grupos de enzimas de cada classe enzimática para o organismo selecionado (

Figura 58).

EC number: 2.7.7.7								
Organism	clusters							
Homo sapiens (human) Eukaryotes Animals Vertebrates Mammals 8	cluster id 6 (7)	cluster id 10 (2)	cluster id 11 (3)	cluster id 12 (1)	cluster id 14 (3)	cluster id 20 (1)	cluster id 33 (1)	cluster id 51 (1)

EC number: 3.1.3.2			
Organism	clusters		
Homo sapiens (human) Eukaryotes Animals Vertebrates Mammals 3	cluster id 122 (1)	cluster id 123 (2)	cluster id 126 (5)

Figura 58. Tela do AnEnDB que exibe os grupos de enzimas das classes de atividade enzimática 2.7.7.7 e 3.1.3.2 para o organismo *H. sapiens*. O número total de seqüências para cada grupo é representado entre parênteses ao lado da identificação do grupo.

Por fim, para obter as seqüências basta clicar na identificação do grupo (Figura 59).

Protein Identification	Fasta Header	Organism	Sequence	PDB
hsa:5425	>hsa:5425 POLD2; polymerase (DNA directed), delta 2, accessory subunit (EC:2.7.7.7)	Homo sapiens (human) Eukaryotes Animals Vertebrates Mammals	MFSEQAQRHLLPPSANNATFARVPVATVNSSQFFLGRSFSRQVHYHATRLIQH RPFLENRAQQHWGSSGVWKKLCELOPEKCCVVGTLFKAFLQPSLKEVSEHLLFPFP RSKYHPDDELVEDELQRKLGTDVSKLVGTGLAVFQSVRDOGKFLVEDYCFADLAP QKAPPFLTDORFVLLVSLGSLGGGGESLSTQLLDVDTVQLGDEGEQSAHVSRVILA GNLLSHSTQSRDSINKAKYTKTKTQMSVENAKMLDELLQLSABVVDVMPGEPDPTNYT LPQQPHPCMFPLATAYSTLQVTPYQATDGVRFGLTSGQNVSDIFRYSMEDHLEILE WTLRVRHISPTAPDILGCVYFYKTDPIFFECPHVYFCGNTPSFSGKIRGPEQDQVLLVT VPDFSATQACLVNLRSLACQPSISFSGFAEDDDLGGGLGP	3E01
hsa:5427	>hsa:5427 POLE2, DPE2; polymerase (DNA directed), epsilon 2, accessory subunit (EC:2.7.7.7)	Homo sapiens (human) Eukaryotes Animals Vertebrates Mammals	MAPERLRSRALSFAKLRGLLRGENKYLTEALQSISELEEDKLEKINAVIEKQPLSNM IEKSDVEAMQEQSQDQVDETLMTNHPAPLFGTPDKAEMFRERYTLHQRTIRHELFT PPVIGSHPDESSGKFLQKLTETLGGTKIGDANVLSHMTQLEKGFLEDFDTQVQLDLS KAQHSGLLYTEACFLVAEGWVEGQVFNANAFPPTEPSSTIRAVYGNINFFGGPSNATSK TSAKLQLEENKDIANFYVLSVDVLDQVELEKLRHIFAGYSPAPPTCTILCGNFSAPYG KNQVQALKDSLKTDICIEYVDHSGSRVFPVPGEDPFGGSLRPLFAESTLNEFRQR VFFSVITTNPCRCQYCTQVTFLEEDVANKMCRNCRFPSSNLFNFHFKLTSQGHLP LPLVYCVVWYDVALRVVFPVOLLVADKYDFPTTNTTECLLNGFSRPSGFSKVPYP SNKTVEDSKLQGF	2V6Z

Figura 59. Tela do AnEnDB exibindo as seqüências do grupo 10 da classe de atividade enzimática 2.7.7.7, em *H. Sapiens*.

No exemplo da Figura 59 as seqüências possuem identificador PDB. Em alguns casos as seqüências são pouco informativas e é necessário caracterizar suas estruturas tridimensionais utilizando outros recursos como modelagem comparativa. Outras caracterizações são importantes para avançar no estudo de analogia intragenômica como o estudo das vias metabólicas envolvidas e os perfis de expressão das formas identificadas.

Validação de Dados

A versão atual do AnEnDB utiliza o KEGG como origem primária de dados. Ao longo do desenvolvimento deste projeto foram descobertas inconsistências em seus dados. Para garantir e conferir a veracidade das informações entre o KEGG e o banco de dados relacional do AnEnDB foram escritos diversos *scripts* na linguagem *Bash* para conferências simples de quantidades de registros.

O *script* *check-total-proteins.sh* retorna o total de proteínas assim como o *check-total-organisms.sh* retorna o total de organismos. O mesmo foi feito para classes de atividades enzimáticas, vias metabólicas etc. Tais *scripts* executam contagens simples e contagens de relacionamentos para posteriormente serem manualmente (via comandos SQL na interface do PostgreSQL) conferidos em relação aos totais encontrados nas tabelas do banco de dados relacional.

Validações mais complexas como, por exemplo, os resultados obtidos com o agrupamento das sequências enzimáticas, por envolverem um volume grande de processamento e por não estarem disponíveis em nenhuma forma pré-processada, foram validados com o uso de resultados da versão anterior do AnEnPi junto com a experiência de pesquisadores que já utilizaram dados sobre grupos de enzimas análogas. Por fim, pesquisadores que estão atualmente estudando enzimas análogas foram convidados para repetirem as buscas que fizeram no início de seus estudos utilizando, agora, o AnEnDB. De um modo geral o uso de *scripts*, experiência e repetição de buscas resultou numa validação positiva, ou seja, não há inconsistências dos dados presentes no banco de dados relacional, sejam eles os dados pré-processados ou dados obtidos através de processamento posterior (agrupamento das sequências). Porém, algumas características do KEGG surgiram indicando erros que precisaram ser resolvidas. Algumas linhas de descrição de proteínas possuíam um parêntese no meio da *string* que descrevia o número EC. Esse fato impedia que algumas proteínas fossem inseridas no banco de dados relacional, pois a expressão regular (48) que definia o formato de um número EC não reconhece *strings* EC com um parêntese no meio. Outras características que afetaram diretamente as classes de processamento de texto foram caracteres como “\” e “/” inseridos dentro da linha de descrição das proteínas. Cada execução dos processadores de texto e posterior identificação de que o número total de proteínas no banco de dados relacional diferia do encontrado nos arquivos FASTA implicava

na intervenção manual para corrigir tais problemas. Mais grave do que a situação das linhas de descrição das proteínas em relação ao número total de proteínas no banco de dados relacional, foi a constatação de que o KEGG possui divergência entre as anotações destas linhas de descrição e outros arquivos que descrevem as proteínas. O mais significativo foi a ocorrência de arquivos que relacionam números EC e organismos não terem a mesma correlação com o que está descrito nas linhas de descrição das proteínas nos arquivos FASTA. O Número EC 5.3.1.6, por exemplo, que apresenta analogia intergenômica entre *T. cruzi* e *H. sapiens*, não está registrado em nenhuma linha de descrição de proteína e sim apenas no arquivo *tcr_enzymes.list*. O total de números ECs que aparecem em arquivos *organismo_enzymes.list* e que não aparecem nas linhas de descrição de proteínas é algo em torno de 1.000. Tal problema afetou diretamente o agrupamento das enzimas em grupos de análogos e implicou que uma primeira versão do AnEnDB apresentasse informações inconsistentes e incompletas. O algoritmo de agrupamento de análogos é o mais custoso em termos computacionais e a correta identificação do problema e posterior solução impactou diretamente no tempo de desenvolvimento do projeto. As inconsistências encontradas reforçaram ainda mais a necessidade de intervenção manual na adição de futuras versões do KEGG e de outros bancos de dados, e reforçaram igualmente a importância de o AnEnDB ser modelado em camadas com responsabilidades bem definidas e restritas.

CONCLUSÃO

O AnEnDB é um *software* e como tal ainda pode ser constantemente modificado para se adaptar a novas demandas, acolher correções, novos recursos e novas maneiras de interação com usuários. Apesar disso, o que foi apresentado neste trabalho demonstra que as etapas já concluídas são suficientes para que o AnEnDB sirva como ferramenta no auxílio em pesquisas sobre enzimas análogas. A simplicidade na maneira em que apresenta os dados confirma que é capaz de poupar tempo de pesquisadores interessados na área.

Diferentes projetos estão sendo desenvolvidos por nosso grupo de Bioinformática do Laboratório de Genômica Funcional e Bioinformática (LAGFB,IOC/Fiocruz) cuja fonte inicial de dados foi o resultado da identificação de enzimas análogas. Um desses trabalhos busca identificar e caracterizar analogia intragenômica em *H. sapiens*, tentando responder à pergunta mencionada anteriormente sobre a possibilidade de formas análogas intragenômicas desempenharem papéis biológicos distintos no organismo que as expressa. Outro projeto igualmente importante busca encontrar alvos moleculares para novos fármacos a partir da análise de enzimas análogas entre *T. cruzi* e humanos (28) como por exemplo as enzimas isoprenil difosfato e fosfo-mevalonato cinase, ambas análogas nesses organismos e pertencentes ao metabolismo de lipídios, fundamentais na constituição de membranas celulares, armazenamento de energia, atuando como cofatores enzimáticos, sinalizadores celulares dentre outras funções importantes.

Sem o AnEnDB os pesquisadores precisam (assim como foi necessário nos estudos citados) criar seus próprios *scripts parsers* e posteriormente contextualizar os dados com outras informações manualmente (ou com mais *scripts parsers*). Dependendo da habilidade, tempo ou experiência do pesquisador, isso pode consumir tempo que poderia estar sendo utilizado para aprofundar mais suas pesquisas..

Os métodos e tecnologias utilizadas comprovaram ser eficazes para um sistema *web* como o proposto e, mais importante, comprovaram ser uma base organizada e consistente para futuros novos recursos e apresentações de mais informações contextualizadas (perfis probabilísticos, mapas gráficos de vias

metabólicas, outros bancos de dados primários, estudos estatísticos, entre outros). O contínuo desenvolvimento do AnEnDB fornecerá não somente mais informações mas também mais conhecimento sobre analogia em atividades enzimáticas.

Portanto, o AnEnDB, através de seu banco de dados único e de seus recursos de programação, permite que instâncias de analogia intergenômica e intragenômica sejam identificadas em organismos com genomas completamente sequenciados, possibilitando pesquisas tanto de cunho básico quanto aplicado às ciências biomédicas.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Mount DW. Bioinformatics: Sequence and Genome Analysis. 2004. 692 p.
2. Luscombe NM, Greenbaum D, Gerstein M. What is bioinformatics? A proposed definition and overview of the field. [Internet]. [cited 2016 Dec 5]. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/11552348>
3. Smith DJ. Applications of bioinformatics and computational biology to influenza surveillance and vaccine strain selection. Vaccine [Internet]. 2003 May 1 [cited 2016 Nov 13];21(16):1758–61. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/12686090>
4. Gupta AK, Kaur K, Rajput A, Dhanda SK, Sehgal M, Khan MS, et al. ZikaVR: An Integrated Zika Virus Resource for Genomics, Proteomics, Phylogenetic and Therapeutic Analysis. Sci Rep [Internet]. 2016 Sep 16 [cited 2016 Nov 13];6:32713. Available from: <http://www.nature.com/articles/srep32713>
5. WHO e-Recruit - Scientist (Bioinformatics). (IARC/13/FT492) [Internet]. [cited 2016 Nov 13]. Available from: https://erecruit.who.int/public/hrd-cl-vac-view.asp?o_c=1000&jobinfo_uid_c=28275&vacIng=en
6. Sanseverino W, Roma G, De Simone M, Faino L, Melito S, Stupka E, et al. PRGdb: a bioinformatics platform for plant resistance gene analysis. Nucleic Acids Res [Internet]. 2010 Jan [cited 2016 Nov 13];38(Database issue):D814-21. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/19906694>
7. Alemu K. The role and application of bioinformatics in plant disease management. Adv Life Sci Technol. 2015;28:28–34.
8. Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, Boldt J, et al. Metabolic engineering of Escherichia coli for direct production of 1,4-butanediol. Nat Chem Biol [Internet]. 2011;7(7):445–52. Available from: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=21602812
9. Mestrado: "Usos Sustentáveis de Recursos Naturais em Regiões Tropicais";

10. Stryer L. Biochemistry. 1995. 1064 p.
11. Tipton K, Boyce S. History of the enzyme nomenclature system. *Bioinformatics*. 2000;16:34–40.
12. EBI. EC 2.7.1.40 description [Internet]. Available from: http://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/enzymes/GetPage.pl?ec_number=2.7.1.40
13. Kornberg H. The study of metabolic pathways [Internet]. Available from: <http://global.britannica.com/science/metabolism/The-study-of-metabolic-pathways>
14. Rodriguez A, Martínez JA, Flores N, Escalante A, Gosset G, Bolivar F. Engineering *Escherichia coli* to overproduce aromatic amino acids and derived compounds. *Microb Cell Fact* [Internet]. 2014;13(1):126. Available from: <http://www.microbialcellfactories.com/content/13/1/126>
15. Center FHM. Metabolic Disorders [Internet]. Available from: <https://www.floridahospital.com/metabolic-disorders>
16. Hatzios SK, Bertozzi CR. The regulation of sulfur metabolism in mycobacterium tuberculosis. *PLoS Pathog*. 2011;7(7):1–8.
17. Marie SKN, Shinjo SMO. Metabolism and brain cancer. *Clinics (Sao Paulo)* [Internet]. 2011;66 Suppl 1:33–43. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/23320861>
18. Fitch WM. Homology. *Trends Genet* [Internet]. 2000;16(5):227–31. Available from: <http://www.sciencedirect.com/science/article/pii/S0168952500020059>
19. Barton. *Evolution*. 1st ed. Cold Spring Harbor Laboratory Press; 2007. 833 p.
20. Magadum S, Banerjee U, Murugan P, Gangapur D, Ravikesavan R. Gene duplication as a major force in evolution. *J Genet* [Internet]. 2013 Apr [cited 2016 Nov 15];92(1):155–61. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/23640422>
21. Koonin E V. An apology for orthologs - or brave new memes. *Genome Biol* [Internet]. 2001 [cited 2016 Nov 15];2(4):COMMENT1005. Available from:

<http://www.ncbi.nlm.nih.gov/pubmed/11305932>

22. Doolittle RF. Convergent evolution: the need to be explicit. *Trends Biochem Sci* [Internet]. 1994 Jan [cited 2016 Nov 15];19(1):15–8. Available from: <http://linkinghub.elsevier.com/retrieve/pii/0968000494901678>
23. Omelchenko M V, Galperin MY, Wolf YI, Koonin E V. Non-homologous isofunctional enzymes: a systematic analysis of alternative solutions in enzyme evolution. *Biol Direct* [Internet]. 2010 Apr 30 [cited 2016 Nov 16];5:31. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/20433725>
24. Gherardini PF, Wass MN, Helmer-Citterich M, Sternberg MJE. Convergent Evolution of Enzyme Active Sites Is not a Rare Phenomenon. *J Mol Biol.* 2007;372(3):817–45.
25. Osterman A. Missing genes in metabolic pathways: a comparative genomics approach. *Curr Opin Chem Biol* [Internet]. 2003 Apr [cited 2016 Nov 16];7(2):238–51. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S1367593103000279>
26. Cordwell SJ. Microbial genomes and “missing” enzymes: redefining biochemical pathways. *Arch Microbiol* [Internet]. 1999 Oct 14 [cited 2016 Nov 16];172(5):269–79. Available from: <http://link.springer.com/10.1007/s002030050780>
27. Galperin MY, Walker DR, Koonin E V. Analogous enzymes: independent inventions in enzyme evolution. *Genome Res* [Internet]. 1998 Aug [cited 2016 Nov 15];8(8):779–90. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/9724324>
28. Alves-Ferreira M, Guimarães ACR, Capriles PV da SZ, Dardenne LE, Degrave WM. A new approach for potential drug target discovery through in silico metabolic pathway analysis using *Trypanosoma cruzi* genome information. *Mem Inst Oswaldo Cruz* [Internet]. 2009 Dec [cited 2016 Nov 16];104(8):1100–10. Available from: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0074-02762009000800006&lng=en&nrm=iso&tlng=en
29. Singh S, Singh G, Gautam B, Farmer R, Jain PA, Yadav PK. Bio-Physics In

- silico metabolic pathway analysis of trichomonas vaginalis for potential drug targets. 2011;32(November):1991–4.
30. Sorokina M, Stam M, Médigue C, Lespinet O, Vallenet D. Profiling the orphan enzymes. *Biol Direct* [Internet]. 2014 Jun 6 [cited 2016 Nov 16];9:10. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24906382>
 31. Zou D, Ma L, Yu J, Zhang Z. Biological databases for human research. *Genomics, Proteomics Bioinforma* [Internet]. 2015;13(1):55–63. Available from: <http://dx.doi.org/10.1016/j.gpb.2015.01.006>
 32. WormBase. WormBase [Internet]. [cited 2016 May 5]. Available from: <http://www.wormbase.org/>
 33. Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. GenBank. *Nucleic Acids Res*. 2014;42(D1):32–7.
 34. Oxford Journals | Science & Mathematics | Nucleic Acids Research [Internet]. [cited 2016 Nov 17]. Available from: <http://nar.oxfordjournals.org/>
 35. Rigden DJ, Fernández-Suárez XM, Galperin MY. The 2016 database issue of nucleic acids research and an updated molecular biology database collection. *Nucleic Acids Res*. 2016;44(D1):D1–6.
 36. MEDLINE. MEDLINE [Internet]. [cited 2016 May 5]. Available from: <https://www.nlm.nih.gov/pubs/factsheets/medline.html>
 37. Wren JD, Bateman A. Databases, data tombs and dust in the wind. *Bioinformatics*. 2008;24(19):2127–8.
 38. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* [Internet]. 1999;28(1):27–30. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/10592173>
 39. KEGG. KEGG 17 Databases [Internet]. [cited 2016 May 5]. Available from: <http://www.genome.jp/kegg/kegg1a.html>
 40. Lo Conte L. SCOP: a Structural Classification of Proteins database. *Nucleic Acids Res* [Internet]. 2000;28(1):257–9. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/28.1.257>

41. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. *Nucleic Acids Res.* 2000;28(1):235–42.
42. PDB. PDB Website [Internet]. [cited 2016 May 5]. Available from: <http://www.rcsb.org/pdb/home/home.do>
43. PDB. PDB REST API [Internet]. [cited 2016 May 5]. Available from: <http://www.rcsb.org/pdb/software/rest.do>
44. Cood EF. A Relational Model of Data for Large Shared Data Banks. *IBM Res Lab Calif* [Internet]. 1969;13(6). Available from: <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
45. Centers for Medicare & Medicaid Services. Selecting a development approach. *Centers Medicare Medicaid Serv* [Internet]. 2008;1–10. Available from: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
46. Sabbagh R. *Scrum - Gestão Ágil Para Projetos de Sucesso*. 2013. 270 p.
47. Beck K. *Extreme Programming Explained: Embrace Change*. 2004. 218 p.
48. Python.org. Regular Expression [Internet]. [cited 2016 May 5]. Available from: <https://docs.python.org/2/howto/regex.html>
49. Otto TD, Guimarães ACR, Degraive WM, de Miranda AB. AnEnPi: identification and annotation of analogous enzymes. *BMC Bioinformatics*. 2008;9:544.
50. Camacho C, Madden T, Ma N, Tao T, Agarwala R, Morgulis A. *BLAST Command Line Applications User Manual, BLAST® Help* [Internet]. Natl Cent Biotechnol Inf (US), Bethesda, MD USA. 2008;
51. Perl.org. Perl.org [Internet]. [cited 2016 May 5]. Available from: <http://www.perl.org>
52. Python.org. Python.org [Internet]. [cited 2016 May 5]. Available from: <http://www.python.org>
53. PostgreSQL.org. PostgreSQL [Internet]. [cited 2016 May 5]. Available from: <http://www.postgresql.org/>
54. OpenSource.org. OpenSource.org [Internet]. [cited 2016 May 5]. Available

- from: <https://opensource.org/>
55. PostgreSQL.org. PostgreSQL Licence [Internet]. [cited 2016 May 5]. Available from: <https://opensource.org/licenses/postgresql>
 56. RFC-Base. RFC HTTP [Internet]. [cited 2016 May 5]. Available from: <http://www.rfc-base.org/rfc-2616.html>
 57. GitHub.org. GitHub.org [Internet]. [cited 2016 May 5]. Available from: <http://github.info/>
 58. Google.com. GoogleTrends [Internet]. [cited 2016 May 5]. Available from: <https://www.google.com.br/trends/?hl=pt-PT>
 59. Tiobe.com. Tiobe.com [Internet]. [cited 2016 May 5]. Available from: http://www.tiobe.com/tiobe_index
 60. Redmonk.org. Redmonk.org [Internet]. [cited 2016 May 5]. Available from: <http://redmonk.com/>
 61. Marcia Cappel. Academia and Programming Language Preferences [Internet]. [cited 2016 May 5]. Available from: <http://redmonk.com/sogrady/2013/04/04/academia-and-programming-languages/>
 62. SQLAlchemy.org. SQLAlchemy.org [Internet]. [cited 2016 May 5]. Available from: <http://www.sqlalchemy.org>
 63. Fowler M. ORMHate [Internet]. [cited 2016 May 5]. Available from: <http://martinfowler.com/bliki/OrmHate.html>
 64. Chaudhary M. Importance of Software Documentation in Software Development [Internet]. [cited 2016 May 5]. Available from: <https://www.linkedin.com/pulse/20140612054919-69055394-importance-of-software-documentation-in-software-development>
 65. Sphinx-doc.org. Sphinx - Python Documentation Generator [Internet]. [cited 2016 May 5]. Available from: <http://www.sphinx-doc.org/>
 66. Sauv e JP. Frameworks [Internet]. [cited 2016 May 5]. Available from: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>

67. Flask. Flask [Internet]. [cited 2016 May 5]. Available from: <http://flask.pocoo.org/>
68. Werkzeug. Werkzeug [Internet]. [cited 2016 May 5]. Available from: <http://werkzeug.pocoo.org/>
69. Jinja. Jinja 2 [Internet]. [cited 2016 May 5]. Available from: <http://jinja.pocoo.org/docs/dev/>
70. WSGI. WSGI [Internet]. [cited 2016 May 5]. Available from: <http://wsgi.readthedocs.io/en/latest/>
71. Rouse M. REST (representational state transfer) [Internet]. [cited 2016 May 5]. Available from: <http://searchsoa.techtarget.com/definition/REST>
72. W3schools.com. HTML Introduction [Internet]. [cited 2016 May 5]. Available from: http://www.w3schools.com/html/html_intro.asp
73. KEGG. KEGG Taxonomic Tree [Internet]. [cited 2016 May 5]. Available from: http://www.genome.jp/kegg/catalog/org_list.html
74. Opperdoes FR. The trypanosomatidae: Amazing organisms. *J Bioenerg Biomembr.* 1994;26(2):145–6.
75. Worldmeters.info. WorldMeters [Internet]. [cited 2016 May 5]. Available from: <http://www.worldometers.info>
76. IBGE. Instituto Brasileiro de Estatística [Internet]. [cited 2016 May 5]. Available from: www.ibge.gov.br
77. Fiocruz. Agência Fiocruz de Notícias [Internet]. [cited 2016 May 5]. Available from: <http://agencia.fiocruz.br/doenca-de-chagas>
1. Mount DW. *Bioinformatics: Sequence and Genome Analysis.* 2004. 692 p.
2. Luscombe NM, Greenbaum D, Gerstein M. What is bioinformatics? A proposed definition and overview of the field. [Internet]. [cited 2016 Dec 5]. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/11552348>
3. Smith DJ. Applications of bioinformatics and computational biology to influenza surveillance and vaccine strain selection. *Vaccine* [Internet]. 2003 May 1 [cited 2016 Nov 13];21(16):1758–61. Available from:

<http://www.ncbi.nlm.nih.gov/pubmed/12686090>

4. Gupta AK, Kaur K, Rajput A, Dhanda SK, Sehgal M, Khan MS, et al. ZikaVR: An Integrated Zika Virus Resource for Genomics, Proteomics, Phylogenetic and Therapeutic Analysis. *Sci Rep* [Internet]. 2016 Sep 16 [cited 2016 Nov 13];6:32713. Available from: <http://www.nature.com/articles/srep32713>
5. WHO e-Recruit - Scientist (Bioinformatics). (IARC/13/FT492) [Internet]. [cited 2016 Nov 13]. Available from: https://erecruit.who.int/public/hrd-cl-vac-view.asp?o_c=1000&jobinfo_uid_c=28275&vacIng=en
6. Sanseverino W, Roma G, De Simone M, Faino L, Melito S, Stupka E, et al. PRGdb: a bioinformatics platform for plant resistance gene analysis. *Nucleic Acids Res* [Internet]. 2010 Jan [cited 2016 Nov 13];38(Database issue):D814-21. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/19906694>
7. Alemu K. The role and application of bioinformatics in plant disease management. *Adv Life Sci Technol*. 2015;28:28–34.
8. Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, Boldt J, et al. Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. *Nat Chem Biol* [Internet]. 2011;7(7):445–52. Available from: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=21602812
9. Mestrado: “Usó Sustentável de Recursos Naturais em Regiões Tropicais”
10. Stryer L. *Biochemistry*. 1995. 1064 p.
11. Tipton K, Boyce S. History of the enzyme nomenclature system. *Bioinformatics*. 2000;16:34–40.
12. EBI. EC 2.7.1.40 description [Internet]. Available from: http://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/enzymes/GetPage.pl?ec_number=2.7.1.40
13. Kornberg H. The study of metabolic pathways [Internet]. Available from: <http://global.britannica.com/science/metabolism/The-study-of-metabolic-pathways>

14. Rodriguez A, Martínez JA, Flores N, Escalante A, Gosset G, Bolivar F. Engineering *Escherichia coli* to overproduce aromatic amino acids and derived compounds. *Microb Cell Fact* [Internet]. 2014;13(1):126. Available from: <http://www.microbialcellfactories.com/content/13/1/126>
15. Center FHM. Metabolic Disorders [Internet]. Available from: <https://www.floridahospital.com/metabolic-disorders>
16. Hatzios SK, Bertozzi CR. The regulation of sulfur metabolism in mycobacterium tuberculosis. *PLoS Pathog*. 2011;7(7):1–8.
17. Marie SKN, Shinjo SMO. Metabolism and brain cancer. *Clinics (Sao Paulo)* [Internet]. 2011;66 Suppl 1:33–43. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/23320861>
18. Fitch WM. Homology. *Trends Genet* [Internet]. 2000;16(5):227–31. Available from: <http://www.sciencedirect.com/science/article/pii/S0168952500020059>
19. Barton. *Evolution*. 1st ed. Cold Spring Harbor Laboratory Press; 2007. 833 p.
20. Magadum S, Banerjee U, Murugan P, Gangapur D, Ravikesavan R. Gene duplication as a major force in evolution. *J Genet* [Internet]. 2013 Apr [cited 2016 Nov 15];92(1):155–61. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/23640422>
21. Koonin E V. An apology for orthologs - or brave new memes. *Genome Biol* [Internet]. 2001 [cited 2016 Nov 15];2(4):COMMENT1005. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/11305932>
22. Doolittle RF. Convergent evolution: the need to be explicit. *Trends Biochem Sci* [Internet]. 1994 Jan [cited 2016 Nov 15];19(1):15–8. Available from: <http://linkinghub.elsevier.com/retrieve/pii/0968000494901678>
23. Omelchenko M V, Galperin MY, Wolf YI, Koonin E V. Non-homologous isofunctional enzymes: a systematic analysis of alternative solutions in enzyme evolution. *Biol Direct* [Internet]. 2010 Apr 30 [cited 2016 Nov 16];5:31. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/20433725>
24. Gherardini PF, Wass MN, Helmer-Citterich M, Sternberg MJE. Convergent Evolution of Enzyme Active Sites Is not a Rare Phenomenon. *J Mol Biol*.

- 2007;372(3):817–45.
25. Osterman A. Missing genes in metabolic pathways: a comparative genomics approach. *Curr Opin Chem Biol* [Internet]. 2003 Apr [cited 2016 Nov 16];7(2):238–51. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S1367593103000279>
 26. Cordwell SJ. Microbial genomes and “missing” enzymes: redefining biochemical pathways. *Arch Microbiol* [Internet]. 1999 Oct 14 [cited 2016 Nov 16];172(5):269–79. Available from: <http://link.springer.com/10.1007/s002030050780>
 27. Galperin MY, Walker DR, Koonin E V. Analogous enzymes: independent inventions in enzyme evolution. *Genome Res* [Internet]. 1998 Aug [cited 2016 Nov 15];8(8):779–90. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/9724324>
 28. Alves-Ferreira M, Guimarães ACR, Capriles PV da SZ, Dardenne LE, Degraeve WM. A new approach for potential drug target discovery through in silico metabolic pathway analysis using *Trypanosoma cruzi* genome information. *Mem Inst Oswaldo Cruz* [Internet]. 2009 Dec [cited 2016 Nov 16];104(8):1100–10. Available from: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0074-02762009000800006&lng=en&nrm=iso&tlng=en
 29. Singh S, Singh G, Gautam B, Farmer R, Jain PA, Yadav PK. Bio-Physics In silico metabolic pathway analysis of *trichomonas vaginalis* for potential drug targets. 2011;32(November):1991–4.
 30. Sorokina M, Stam M, Médigue C, Lespinet O, Vallenet D. Profiling the orphan enzymes. *Biol Direct* [Internet]. 2014 Jun 6 [cited 2016 Nov 16];9:10. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24906382>
 31. Zou D, Ma L, Yu J, Zhang Z. Biological databases for human research. *Genomics, Proteomics Bioinforma* [Internet]. 2015;13(1):55–63. Available from: <http://dx.doi.org/10.1016/j.gpb.2015.01.006>
 32. WormBase. WormBase [Internet]. [cited 2016 May 5]. Available from: <http://www.wormbase.org/>

33. Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. GenBank. *Nucleic Acids Res.* 2014;42(D1):32–7.
34. Oxford Journals | Science & Mathematics | Nucleic Acids Research [Internet]. [cited 2016 Nov 17]. Available from: <http://nar.oxfordjournals.org/>
35. Rigden DJ, Fernández-Suárez XM, Galperin MY. The 2016 database issue of nucleic acids research and an updated molecular biology database collection. *Nucleic Acids Res.* 2016;44(D1):D1–6.
36. MEDLINE. MEDLINE [Internet]. [cited 2016 May 5]. Available from: <https://www.nlm.nih.gov/pubs/factsheets/medline.html>
37. Wren JD, Bateman A. Databases, data tombs and dust in the wind. *Bioinformatics.* 2008;24(19):2127–8.
38. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* [Internet]. 1999;28(1):27–30. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/10592173>
39. KEGG. KEGG 17 Databases [Internet]. [cited 2016 May 5]. Available from: <http://www.genome.jp/kegg/kegg1a.html>
40. Lo Conte L. SCOP: a Structural Classification of Proteins database. *Nucleic Acids Res* [Internet]. 2000;28(1):257–9. Available from: <http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/28.1.257>
41. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. *Nucleic Acids Res.* 2000;28(1):235–42.
42. PDB. PDB Website [Internet]. [cited 2016 May 5]. Available from: <http://www.rcsb.org/pdb/home/home.do>
43. PDB. PDB REST API [Internet]. [cited 2016 May 5]. Available from: <http://www.rcsb.org/pdb/software/rest.do>
44. Cood EF. A Relational Model of Data for Large Shared Data Banks. *IBM Res Lab Calif* [Internet]. 1969;13(6). Available from: <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
45. Centers for Medicare & Medicaid Services. Selecting a development approach.

- Centers Medicare Medicaid Serv [Internet]. 2008;1–10. Available from: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
46. Sabbagh R. Scrum - Gestão Ágil Para Projetos de Sucesso. 2013. 270 p.
 47. Beck K. Extreme Programming Explained: Embrace Change. 2004. 218 p.
 48. Python.org. Regular Expression [Internet]. [cited 2016 May 5]. Available from: <https://docs.python.org/2/howto/regex.html>
 49. Otto TD, Guimarães ACR, Degraive WM, de Miranda AB. AnEnPi: identification and annotation of analogous enzymes. BMC Bioinformatics. 2008;9:544.
 50. Camacho C, Madden T, Ma N, Tao T, Agarwala R, Morgulis A. BLAST Command Line Applications User Manual, BLAST® Help [Internet]. Natl Cent Biotechnol Inf (US), Bethesda, MD USA. 2008;
 51. Perl.org. Perl.org [Internet]. [cited 2016 May 5]. Available from: <http://www.perl.org>
 52. Python.org. Python.org [Internet]. [cited 2016 May 5]. Available from: <http://www.python.org>
 53. PostgreSQL.org. PostgreSQL [Internet]. [cited 2016 May 5]. Available from: <http://www.postgresql.org/>
 54. OpenSource.org. OpenSource.org [Internet]. [cited 2016 May 5]. Available from: <https://opensource.org/>
 55. PostgreSQL.org. PostgreSQL Licence [Internet]. [cited 2016 May 5]. Available from: <https://opensource.org/licenses/postgresql>
 56. RFC-Base. RFC HTTP [Internet]. [cited 2016 May 5]. Available from: <http://www.rfc-base.org/rfc-2616.html>
 57. GitHut.org. GitHut.org [Internet]. [cited 2016 May 5]. Available from: <http://github.info/>
 58. Google.com. GoogleTrends [Internet]. [cited 2016 May 5]. Available from: <https://www.google.com.br/trends/?hl=pt-PT>

59. Tiobe.com. Tiobe.com [Internet]. [cited 2016 May 5]. Available from: http://www.tiobe.com/tiobe_index
60. Redmonk.org. Redmonk.org [Internet]. [cited 2016 May 5]. Available from: <http://redmonk.com/>
61. Marcia Cappel. Academia and Programming Language Preferences [Internet]. [cited 2016 May 5]. Available from: <http://redmonk.com/sograde/2013/04/04/academia-and-programming-languages/>
62. SQLAlchemy.org. SQLAlchemy.org [Internet]. [cited 2016 May 5]. Available from: <http://www.sqlalchemy.org>
63. Fowler M. ORMHate [Internet]. [cited 2016 May 5]. Available from: <http://martinfowler.com/bliki/OrmHate.html>
64. Chaudhary M. Importance of Software Documentation in Software Development [Internet]. [cited 2016 May 5]. Available from: <https://www.linkedin.com/pulse/20140612054919-69055394-importance-of-software-documentation-in-software-development>
65. Sphinx-doc.org. Sphinx - Python Documentation Generator [Internet]. [cited 2016 May 5]. Available from: <http://www.sphinx-doc.org/>
66. Sauv e JP. Frameworks [Internet]. [cited 2016 May 5]. Available from: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>
67. Flask. Flask [Internet]. [cited 2016 May 5]. Available from: <http://flask.pocoo.org/>
68. Werkzeug. Werkzeug [Internet]. [cited 2016 May 5]. Available from: <http://werkzeug.pocoo.org/>
69. Jinja. Jinja 2 [Internet]. [cited 2016 May 5]. Available from: <http://jinja.pocoo.org/docs/dev/>
70. WSGI. WSGI [Internet]. [cited 2016 May 5]. Available from: <http://wsgi.readthedocs.io/en/latest/>
71. Rouse M. REST (representational state transfer) [Internet]. [cited 2016 May 5].

Available from: <http://searchsoa.techtarget.com/definition/REST>

72. W3schools.com. HTML Introduction [Internet]. [cited 2016 May 5]. Available from: http://www.w3schools.com/html/html_intro.asp
73. KEGG. KEGG Taxonomic Tree [Internet]. [cited 2016 May 5]. Available from: http://www.genome.jp/kegg/catalog/org_list.html
74. Opperdoes FR. The trypanosomatidae: Amazing organisms. *J Bioenerg Biomembr.* 1994;26(2):145–6.
75. Worldmeters.info. WorldMeters [Internet]. [cited 2016 May 5]. Available from: <http://www.worldometers.info>
76. IBGE. Instituto Brasileiro de Estatística [Internet]. [cited 2016 May 5]. Available from: www.ibge.gov.br
77. Fiocruz. Agência Fiocruz de Notícias [Internet]. [cited 2016 May 5]. Available from: <http://agencia.fiocruz.br/doenca-de-chagas>