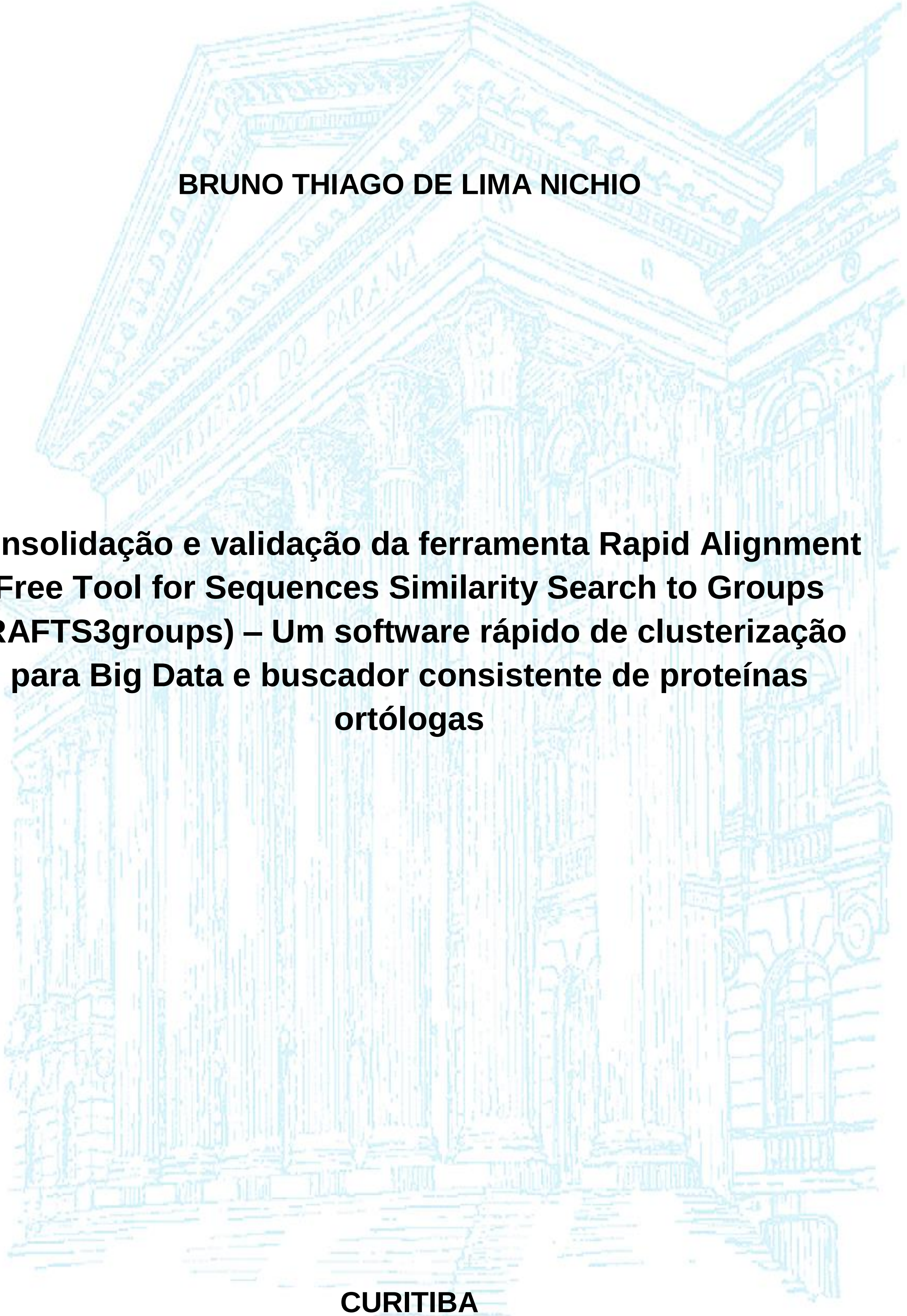
UNIVERSIDADE FEDERAL DO PARANÁ



BRUNO THIAGO DE LIMA NICHIO

# Consolidação e validação da ferramenta Rapid Alignment Free Tool for Sequences Similarity Search to Groups (RAFTS3groups) – Um software rápido de clusterização para Big Data e buscador consistente de proteínas ortólogas

CURITIBA

2016

UNIVERSIDADE FEDERAL DO PARANÁ

BRUNO THIAGO DE LIMA NICHIO

# CONSOLIDAÇÃO E VALIDAÇÃO DA FERRAMENTA *RAPID ALIGNMENT FREE TOOL FOR SEQUENCES SIMILARITY SEARCH TO GROUPS (RAFTS3GROUPS)* – UM SOFTWARE RÁPIDO DE CLUSTERIZAÇÃO PARA *BIG DATA* E BUSCADOR CONSISTENTE DE PROTEÍNAS ORTÓLOGAS

CURITIBA, 2016

BRUNO THIAGO DE LIMA NICHIO

# CONSOLIDAÇÃO E VALIDAÇÃO DA FERRAMENTA *RAPID ALIGNMENT FREE TOOL FOR SEQUENCES SIMILARITY SEARCH TO GROUPS (RAFTS3GROUPS)* – UM SOFTWARE RÁPIDO DE CLUSTERIZAÇÃO PARA *BIG DATA* E BUSCADOR CONSISTENTE DE PROTEÍNAS ORTÓLOGAS

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, área de concentração em Bioinformática.

Orientador: Dr. Roberto Tadeu Raittz
Coorientadores: Dra. Jeroniza Nunes Marchaukoski e Dr. Vinicius Almir Weiss

CURITIBA, 2016

Nichio, Bruno Thiago de Lima
N594 Consolidação e validação da ferramenta rapid alignment free tool for sequences similarity search to groups (rafts3groups) – um software rápido de clusterização para big data e buscador consistente de proteínas ortólogas. - Curitiba, 2016.
89 f.; il.: tab.

Orientador: Roberto Tadeu Raittz
Coorientadores: Jeroniza Nunes Marchaukoski
Vinícius Almir Weiss
Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Educação Profissional e Tecnológica, Curso de Pós-Graduação em Bioinformática.
Inclui Bibliografia.

1. Homologia. 2. Análise por conglomerados. 3. Sistema de recuperação da informação. 4. Software – Desenvolvimento. 5. Bioinformática. I. Raittz, Roberto Tadeu. II. Marchaukoski, Nunes. III. Weiss, Vinícius Amir. IV. Título. V. Universidade Federal do Paraná.

CDD 005.369

## TERMO DE APROVAÇÃO

BRUNO THIAGO DE LIMA NICHIO

**“Consolidação e validação da ferramenta *Rapid Alignment Free Tool for Sequences Similarity Search to Groups* (RAFTS3groups) – um software rápido de clusterização para *Big Data* e buscador consistente de proteínas ortólogas”**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Bioinformática, pelo Programa de Pós-graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná, pela seguinte banca examinadora:

Dr. Roberto Tadeu Raittz
Universidade Federal do Paraná - UFPR

Dr. Helisson Faoro
Fundação Oswaldo Cruz – Instituto Carlos Chagas – FioCruz/PR

Dr. Luiz Ermindo Cavallet
Fac. Est. de Filosofia Ciências e Letras de Paranaguá - UNESPAR

Curitiba, 16 de setembro de 2016

**DEDICATÓRIA**

Primeiramente a Deus,
Aos meus guias e mestres,
A toda a família.

# AGRADECIMENTOS

A vida é um longo processo de aprendizado em que, a cada dia, temos que superar diversos obstáculos, sofrer alguns tropeços, mas, ao final de cada etapa, nos deparamos com uma alvorada de realizações. A cada etapa vencida, sempre há em nossas mentes uma fonte de inspiração, algo que nos faz crer que podemos fazer mais e mais por nós mesmos, por nossos entes mais queridos ou até mesmo desconhecidos.

Agradeço de coração ao meu orientador e grande professor Dr Roberto Tadeu Raitzz, pela tamanha paciência, amizade, ensinamentos e, principalmente, por ter depositado toda sua confiança e energia à minha orientação desde o início dos nossos trabalhos. Agradeço também aos meus queridos co-orientadores Dra Jeroniza Marchauckoski por ser mais que minha educadora, mas ser uma pessoa sempre presente em todos os dias contribuindo e ao Dr Vinícius Almir Weiss que também teve sua brilhante orientação sempre nos momentos mais precisos e, todos citados, honrando-me com suas contribuições e me aperfeiçoando como aluno e também como pessoa.

Deixo minha enorme gratidão a todos os meus amigos e colegas de laboratório, em especial aos meus dois colegas e verdadeiros amigos, o mestrando Alexandre Quadros Lejambre e a mestranda Roxana Beatriz Ribeiro Chaves. Também deposito minha grande gratidão aos meus companheiros, a mestranda Camila Reginatto de Pierri, ao mestrando Aryel Marlus R. de Oliveira, e ao IC Marilson Reque que sempre contribuíram não só com mais conhecimento, mas com toda amizade.

Agradeço, desde o primeiro instante, a todo o programa de Pós-Graduação em Bioinformática e a CAPES, por investirem na formação profissional de cada um de nós e nos permitirem chegarmos até o fim desse processo. Agradeço à Secretaria da Bioinformática, em especial, a Suzana e a Léa que estiveram presentes sempre, nos auxiliando com extrema competência. Agradeço a todos os professores que participaram do programa e que depositaram seus ensinamentos cada um à sua maneira, brilhantemente, crucial à minha formação.

Aos meus familiares e amigos, mesmo que distantes fisicamente, com todo amor e carinho estiveram comigo e, primeiramente, agradeço com toda minha fé e carinho a Deus. A todos que, mais uma vez, direta ou indiretamente, contribuíram com esses curtos, porém valiosíssimos, dois anos de formação profissional e pessoal.

*“O amor é a força mais sutil do mundo”*

Mahatma Gandhi

# RESUMO

Uma das principais análises envolvendo sequências biológicas, imprescindíveis e complexas, é a análise de homologia. A necessidade de desenvolver técnicas e ferramentas computacionais que consigam predizer com mais eficiência grupos de ortólogos e, ao mesmo tempo, lidar com grande volume de informações biológicas, ainda é um grande gargalo a ser superado pela bioinformática. Atualmente, não existe uma única ferramenta eficiente na detecção desses grupos, pois ainda requerem muito esforço computacional e tempo. Metodologias já consolidadas, como o BLAST 'todos contra todos', RBH e ferramentas como o OrthoMCL, demandam um alto custo computacional e falham quando há ortologia, necessitando de uma intervenção manual sofisticada. Diante desse cenário, neste trabalho, aprensentamos um breve *review* referente às técnicas, desenvolvidas entre 2011 até metade de 2017, para a detecção de ortólogos, descrevendo 12 ferramentas e contextualizando os principais problemas ainda a serem superados. A maioria das ferramentas utiliza o algoritmo BLAST como algoritmo padrão predição de homologia entre sequências. Apresentamos também uma nova abordagem para a clusterização de homólogos, a ferramenta RAFTS3groups. Para validarmos a ferramenta utilizamos como base de dados o UniProtKB/Swiss-Prot com outras ferramentas de clusterização o UCLUST e CD-HIT. RAFTS3groups mostrou-se ser mais de 4 vezes mais rápido que o CD-HIT e equiparável em volume de *clusters* e de tempo à ferramenta UCLUST. Para análise e consolidação de homologia, introduzimos uma nova aplicação auxiliar à ferramenta RAFTS3groups, na clusterização de ortólogos, o *script DivideCluster*. Comparamos com o método BLAST 'todos contra todos', analisando 9 genomas completos de *Herbaspirillum* spp. disponíveis no NCBI *genbank*. RAFTS3groups mostrou-se tão eficiente quanto o método, apresentando cerca de 96% de correlação entre os resultados de clusterização de core e pan genoma obtidos.

Palavras-chave: *homologia*, clusterização, *alignment-free*, *k-means*, RAFTS3.

# ABSTRACT

One of the main tests involving biological sequences, essential and complex, is the analysis of homology. The study of homologous genes involved in processes such as cell cycle, DNA repair in simpler organisms, even with large evolutionary distance, there are genes that are shared between primates, yeasts and bacteria, which we call (core-genome). The need to develop computational tools and techniques that can predict more efficiently ortologs groups and handle large volume of biological information is still a problem to be resolved by Bioinformatics. We don't have a single powerful tool in detecting groups that still require a lot of effort and computing time. Tools, already consolidated, as the BLAST ' 'all-against-all' ', RBH, OrthoMCL, demand a high computational cost and fail when there is orthology, requiring manual intervention. In this scenario, in this work we presents a brief review on main techniques, developed between 2011 until early 2016, for the detection of orthologs groups, describing 12 tools and being developed currently and the main problems main problems still to be overcome. We note that most tools uses the BLAST as default prediction of homology between sequences. We also present a new approach for the analysis of homology, the RAFTS3groups tool. We use as the database UniProtKB /Swiss-Prot with the clustering tools the UCLUST and the CD-HIT. RAFTS3groups proved to be more than 4 times faster than CD-HIT and comparable in volume to *clusters* and time with UCLUST tool. In Homology analysis we introduced a new clustering strategy of orthology, the *DivideCluster* algorithm aplication built into the RAFTS3groups. Compared with the BLAST 'all-against-all', analyzing 9 complete genomes from *Herbaspirillum* spp. available by NCBI genbank. RAFTS3groups was shown to be as efficient as the method, showing approximately 96% of the correlation among the clustering results of core and pan genome obtained.

Key-words: homology, clustering, alignment-free, k-means clustering, RAFTS3.

## LISTA DE FIGURAS

# LISTA DE TABELAS

## LISTA DE ABREVIATURAS

BCOM - *Binary Co-Occurrence Matrix*

BD – Banco de Dados

BLAST - *Basic Local Alignment Search Tool*

CD-HIT – *Cluster Database at High Identity with Tolerance*

COG - *Clusters of Orthologous Groups*

COCO-CL - *Correlation Coefficient-based Clustering*

DNA – *DeoxyriboNucleic Acid*

EMBL-EBI - Instituto Europeu de Bioinformática

GHz – *Giga Hertz*

Inparanoid - Automatic Clustering of Orthologs and In-paralogs

KOG - *euKaryotic Orthologous Groups*

LTS - *Long Term Support*

NCBI - *National Center for Biotechnology Information*

OrthoDB - *Ortholog Data Bank*

OrthoMCL - *Markov Cluster algorithm for grouping proteins into multi-species orthologous groups*

PIR - Recurso de Informações de Proteínas

RAFTS3 - *Rapid Alignment Free Tool for Sequences Similarity Search*

RAM – *Random Access Memory*

RBH - *Reciprocal Best Hits*

SIB - Instituto Suiço de Bioinformática

TrEMBL – *Translated EMBL Nucleotide Sequence Data Library*

UniParc - *UniProt Archive*

UniProt - *Universal Protein Resource*

UniProtKB - *UniProt Knowledgebase*

UniRef - *UniProt Reference Clusters*

## SUMÁRIO

## 1 INTRODUÇÃO

Desde o surgimento do sequenciamento genômico em larga escala, a partir de 2002, vêm ganhando força, principalmente nas últimas décadas, estudos e análises de genomas e de proteomas (EMMS et al, 2015). Observa-se que, a partir dessa época, há um aumento exponencial de mais e mais sequências geradas pelo sequenciamento em larga escala e a necessidade da criação de grandes bancos de dados para armazenarem tais informações, o que chamamos de *Big Data* (EMMS et al, 2015). Tal crescimento, é responsável por trazer um grande gargalo ao campo de análise de sequências em bioinformática: a necessidade do desenvolvimento de técnicas e de ferramentas computacionais que consigam lidar com grande volume de informações que utilizem similaridade entre sequencias. Similaridade significativa é uma forte evidência de que, duas sequências ou mais sequencias, são relacionadas por evolução divergente, compartilhando um ancestral comum, o que chamamos de homologia (KOONIN,2005).

O estudo de homologia é também a razão de uso do processo computacional, lógico ou estatístico na detectção de ortólogos, que está estreitamente relacionado com análise comparativa entre genomas e com o dinamismo genômico entre diferentes organismos (KIM et al, 2011). Também é um campo de estudo extremamente importante para a melhoria da anotação funcional de vários organismos (KIM et al, 2011). O campo ainda é destaque em várias análises que ajudam a elucidar processos evolutivos ao longo do surgimento das espécies (WANG et al, 2015).

Desde os primeiros estudos envolvendo a criação de técnicas para inferência de ortologia, a principal dificuldade tem sido a falta de uma metodologia ou de uma ferramenta que fosse eficiente na construção de conjunto de dados (clusterização) de ortólogos. Somente em 2007, surgiu o primeiro estudo referente a análise de ferramentas computacionais envolvendo sensibilidade, acurácia e desempenho de metodologias na detecção de grupos de ortólogos (ALTHENHOFF & DESSIMOZ, 2009) e, com isso, consagrando metodologias “padrões ouro”, como as adaptaçoes dos modelos de *Markov Cluster Algoritm*, adaptaçoes do algoritmo *Basic Local Alignment Search Tool* (BLAST), *Reciprocal Best Hits* (RBH), *Correlation Coefficient-based Clustering* (COCO-CL), *Automatic Clustering of Orthologs and In-paralogs* (Inparanoid). Isso possibilitou o surgimento de várias ferramentas, como o *pipeline integrates a Markov Cluster algorithm for grouping proteins into multi-species* *orthologous groups* (OrthoMCL), que foram sendo consolidadas em estudos subsequentes (KRISTENSEN et al, 2001). O resultado gerou em outras várias publicações e na criação de grandes bancos de dados biológicos contendo *clusters* de Ortólogos como *Clusters of Orthologous Groups/euKaryotic Orthologous Groups* (COG/KOG), *Ortholog Data Bank* (OrthoDB) e o *evolutionary genealogy of genes: Non-supervised* Orthologous *Groups* (eggNOG) (KUZNIAR et al, 2008). Entretanto, a predição de grupos de ortólogos é dificil e, atualmente, não se tem uma única ferramenta eficiente na consolidação de grupos, mas sim um conjunto de ferramentas que atendam a determinadas demandas computacionais e de interesse de cada usuário (ALTHENHOFF & DESSIMOZ, 2009). Também observa-se a necessidade de muitas melhorias ainda a serem feitas para uma predição de ortólogos mais eficiente por parte dessas ferramentas (ALTHENHOFF & DESSIMOZ, 2009). Além do problema de construir relacionamentos entre genomas de Ortólogos, a atualização dos dados armazenados também requer muito esforço computacional e de muito tempo, além de que, muitas vezes, relacionar ortologia entre organismos de parentescos distantes, continua a ser um grande desafio (CHEN & WU, 2010). Isto exige ferramentas de softwares mais eficientes (CURTIS et al, 2013), fato que, ferramentas como BLAST e Inparanoid, falham quando há ortologia, porém o nível de conservação entre Ortólogos é baixo, sendo necessário intervenção manual sofisticada, o que acaba dificultando a automatização do processo de maneira geral (WAGNER et al, 2014). O trabalho é agravado quando, no estudo, precisa-se incluir um número grande de sequências a serem analisadas para inferência de ortologia (BITARD-FEILDEL et al, 2015). Outro problema encontrado é o requerimento de um alto nível de conhecimento de programação por parte dos pesquisadores para analisar grandes volumes de dados e o número cada vez mais crescente de genomas que são depositados nos grande bancos de sequências biológicas que podem ser comparados ao mesmo tempo, o que dificulta a fluidez dos trabalhos (LECHNER et al, 2011). Algumas metodologias e ferramentas, já consolidadas, como o BLAST todos contra todos, RBH, OrthoMCL, demandam um alto custo computacional (LINARD, 2011) o que vai além das capacidades de hardware padrões atuais e requerem acesso a recursos de supercomputadores (LECHNER et al, 2011).

Diante de tudo isso, surge a necessidade de ferramentas que melhorem a sensibilidade na detecção de grupos de ortólogos, mais rápidas e que consilam lidar com grande volume de informações (EMMS & KELLY, 2015). Portanto, em nosso trabalho, objetivamos trazer à comunidade, em forma de um *review*, uma lista de softwares mais recentes, entre 2010 até início de 2016, e também uma proposta alternativa de ferramenta de clusterização, o RAFTS3group, para identificar e consolidar grupos de ortólogos de forma fácil ao pesquisador e livre de algoritmos de alinhamentos como o BLAST, afim de minimizar gastos computacionais e otimizar tempo, disponibilizada de forma gratuita e tão eficiente quanto outras metodologias já consolidadas, conseguindo lidar com grandes volumes de dados biológicos.

# 2 FUNDAMENTAÇÃO TEÓRICA

## 2.1- O ESTUDO DA HOMOLOGIA

Uma das principais análises envolvendo sequências biológicas e complexas é a análise de homologia. Homologia é o relacionamento em que dois organismos descendem, geralmente com divergência, de um organismo ancestral comum (Fitch, 2000). Como em estruturas anatômicas, homologia entre sequências de DNA ou proteína entre organismos diferentes, podem compartilhar de uma ancestralidade comum. Dois segmentos de DNA podem ter compartilhado ascendência por causa de um evento de especiação (ortólogos) ou um evento de duplicação (parálogos) (KOONIN,2005). Homologia entre proteínas ou DNA normalmente é inferida da similaridade entre suas sequências. Alinhamentos de sequências múltiplas são usados para indicar quais as regiões de cada sequência são homólogas. A determinação de ortologia ou paralogia está relacionada a eventos de evolução gênica (Fitch, 2000). Genes que tenham sido duplicados dentro de uma mesma linhagem (linhas horizontais) são parálogos, não importando se possuem a mesma função ou não. Já os genes que foram alterados dentro de linhagens específicas, após especiação (aqueles nos quais, se voltarmos à sua origem, chegamos a uma bifurcação ou Y invertido) são os chamados ortólogos (JENSEN,2001) (FIGURA 1).

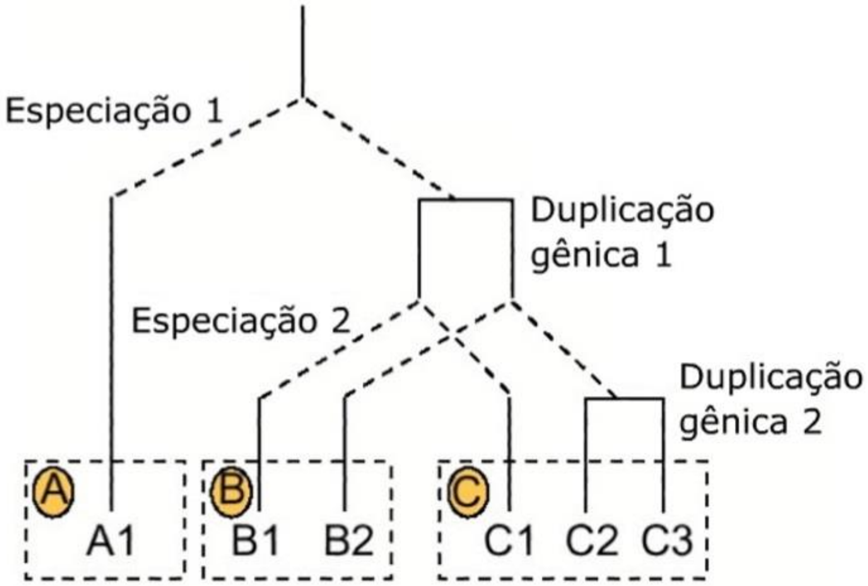


FIGURA 1. DIAGRAMA SIMPLIFICADO DE HOMOLOGIA

Diagrama simplificado de homologia e os subtipos ortologia ( na figura, Especiação 1 e 2) e paralogia (na figura, Duplicação 1 e 2). A, B e C são espécies diferentes. (Adaptada de JENSEN, 2001)

A homologia pode parecer uma concepção abstrata, mas é o princípio orientador de muitas pesquisas biomédicas, como por exemplo, na orientação da escolha de algum organismo modelo, fundamentados pelo grau de homologia exigido para estudar um processo específico ou uma doença (FREEMAN & HERRON, 2009). Sendo mais simplista: o estudo de genes homólogos envolvidos em processos como ciclo celular, de reparo de DNA mesmo em organismos mais simples, mesmo havendo grande distância evolutiva, não exclui a existência de genes que são compartilhados entre primatas, leveduras e bactérias, o que denominamos de *core*-genoma (FREEMAN & HERRON, 2009).

## 2.2 FERRAMENTAS DE CLUSTERIZAÇÃO E CONTRUÇÃO DO *REVIEW*

As ferramentas que lidam com homologia, não importando o subtipo, são ainda destaques no meio científico (EMMS & KELLY, 2015). Devido à importância e o crescimento de vários estudos no desenvolvimento de novas técnicas de Ortologia, desde 2014, realizamos vários levantamentos de metodologias e de ferramentas computacionais, entre 2010 até início de 2016, na predição de proteínas ou de genes

Ortólogos nos mais diversos níveis de estudo com sequências biológicas. A partir de 2010 pois, após esse ano, foi observado um crescente surgimento de novas ferramentas computacionais através de publicações até o presente momento. Foram filtradas informações de vários bancos de publicações científicas, principalmente oriundas do banco *Google Schoolar*, citações de literatura biomédica do MEDLINE utilizando a ferramenta PubMed (NCBI) e pelo portal de recursos de bioinformática do SIB que fornece acesso a ferramentas de software e bases de dados científicos (ExPASy) para conseguirmos encontrar ferramentas, softwares ou pacotes, que possuíssem, como potencial, solucionar a questão da criação e consolidação de grupos de Ortólogos e que também visassem solucionar problemas remanescentes pelas metodologias e ferramentas mais antigas, como, por exemplo, a alta demanda computacional e uso de supermáquinas para lidar com processos na busca por Ortólogos. Cada artigo foi curado manualmente, analisando conteúdos de *abstract* e a revista científica referida, totalizando cerca de 97 artigos só para a revisão. Muitos artigos recolhidos são pertencentes à BMC *bioinformatics*, *Oxford Journals* sessão bioinformática, *Evolutionary Bioinformatics* e *Genome Biology*. Encontramos vários artigos que destacaram ferramentas que se mostraram promissoras por incorporarem novas metodologias ou de adaptar as já consolidadas. Algumas, inclusive, dispõem-se de algoritmos específicos para problemas também específicos envolvendo ortologia entre organismos, tendo em vista que são muitos.

Elencaremos (na seção de Artigos) algumas das principais ferramentas, desenvolvidas ou ainda em desenvolvimento, destacando suas principais características, algoritmos envolvidos, possíveis soluções na tentativa de minimizar problemas específicos encontrados na predição de Ortólogos e, após as análises, direcionar aquelas que realmente se destacaram nesse cenário.

## 2.3 CLUSTERIZAÇÃO E ANÁLISE DE HOMOLOGIA

A clusterização é uma estratégia, em mineração de dados, que visa agrupar sequências de acordo com a similaridade (PROSDOCIMI, 2007). Ela facilita a visualização do relacionamento entre organismos, a conversação entre máquina e banco de dados e a classificação ou categorização de sequências similares (PROSDOCIMI, 2007). Por meio da clusterização, é possível analisar “hierarquias” entre cada sequência gerada no grupo, sendo, portanto, um tipo de modelo de classificação Não-supervisionado, ou seja, sem informação de classe inicial ao processo de clusterização.

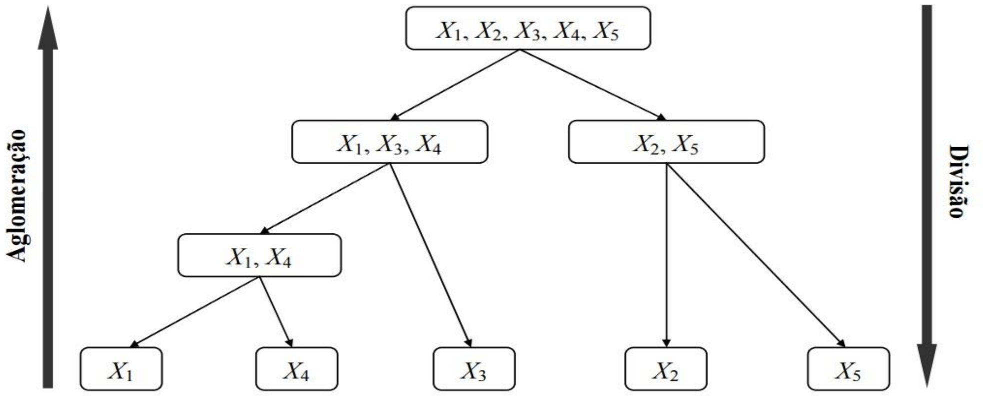


FIGURA 2. EXEMPLO DE ÁRVORE DE *CLUSTERS* NA CLUSTERIZAÇÃO HIERARQUICA.
Exemplificação de conjunto de dados $X_{1,}$ $X_2$, $X_3$, $X_4$ e $X_5$ sendo clusterizados em sentido de agrupar (Aglomeração) ou de separar (Divisão) o conjunto inicial de acordo com afinidade ou não dentro do grupo de origem (OCHI et al, 2010).

Para análises genômicas, infere-se ortologia sequências com pelo menos 30% de identidade ao longo de pelo menos 60% da sua extensão. Abaixo são classificados como não relacionados e muito elevados, próximo a 100%, como cópias. (ZAHA, & PASSAGLIA, 2014).

Considera-se como nota de corte *clusters* com 20% de identidade e 28% de *bit-score* em análises envolvendo alinhamento com peptídeos (RANGEL et al, 2010).

## 2.4 FERRAMENTAS DE CLUSTERIZAÇÃO E DE ANÁLISE

### 2.3.1 RAFTS3groups e o algoritmo *DivideCluster*

A tarefa de encontrar genes homólogos a uma, ou várias, sequências de interesse (*query*) em um banco de dados contendo muitas outras sequências (*subject*), pode ser definida como a obtenção do melhor alinhamento possível da busca contra todos os alvos, marcando cada um destes alinhamentos e escolhendo aqueles cujo *score* superarem um determinado limiar (MAFRA, 2012 não publicado). Infelizmente, a maioria das ferramentas disponíveis utilizam de cálculos de alinhamento, que exigem, além de muito processamento computacional, de muito tempo para análise (LINARD, 2011).

De forma fácil ao utilizador, afim de minimizar tempo e mantendo a consistência na análise de dados com proteínas ortólogas, temos como proposta a ferramenta *Rapid Alignment Free Tool for Sequences Similarity Search to Groups* (RAFTS3groups). Como o próprio nome já diz, a ferramenta não utiliza cálculos de alinhamento, pois é uma ferramenta *alignment free* baseado no algoritmo *Rapid Alignment Free Tool for Sequences Similarity Search* (RAFTS3) (VIALLE, 2013). RAFTS3 (dispoível em https://sourceforge.net/projects/rafts3/) é, de forma sucinta, uma ferramenta que busca similaridade entre sequências de proteínas e que utiliza também um filtro (função *Hash*) para a seleção de candidadtos com base em *k-mers* compartilhados e uma medida de comparação usando uma matriz de co-ocorrencia de resíduos de aminoácidos (BCOM).

RAFTS3groups, inicialmente foi desenvolvido como uma aplicação da ferramenta RAFTS3 para o agrupamento de sequências homólogas e foi descrita anteriormente no trabalho de Coimbra (COIMBRA, 2015). RAFTS3groups recebe como entrada o arquivo multiFASTA de proteínas ou de nucleotídeos a serem agrupados. O *script* cria um banco ao qual é feita a consulta pelo algoritmo RAFTS3. Cada proteína no arquivo de entrada é avaliada, inicialmente, como um possível grupo e o resultado da consulta ao banco avalia a similaridade entre as proteínas pelo valor de *self-score*. Os grupos de ortólogos são formados com base na verificação do número de ocorrências reconhecidas pelo mínimo de similaridade avaliada (por exemplo, *self-score* igual a 0.5 irão gerar vários grupos com similaridade mínima de 50% entre elas) (COIMBRA, 2015).

O funcionamento da ferramenta na predição de ortologia dá-se em duas etapas principais: a primeira instância formata o banco e gera *clusters* de proteínas utilizando algoritmos RAFTS3 e a segunda etapa é a de filtragem e de melhoramento de *clusters* através da análise dos máximos dos mínimos em que, informações de grupo e de organismos, utilizando o cálculo k-*means*, são utilizados para melhor predizer os *clusters* de proteínas com base em ortologia, pela a implementação do algoritmo *DivideCluster*.

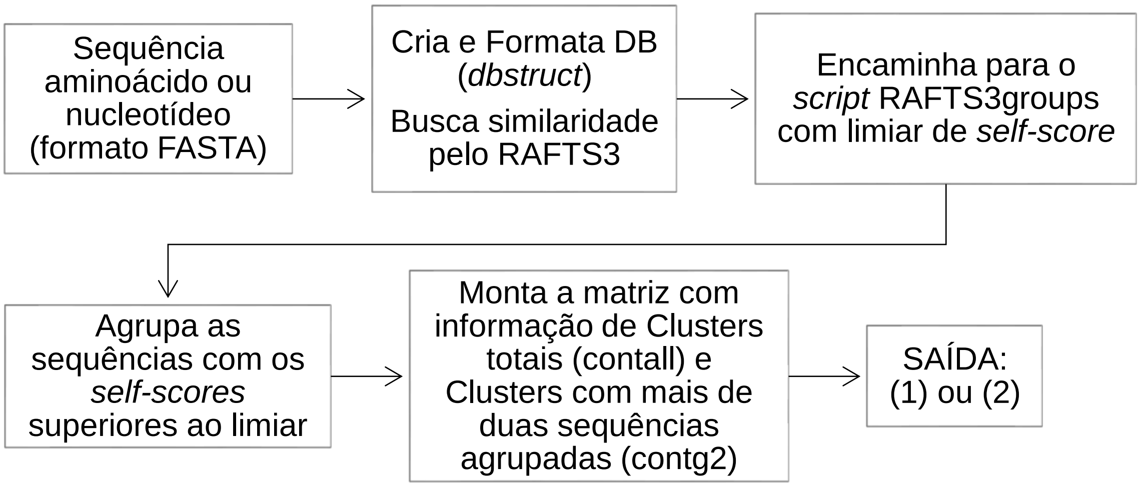

Sequência aminoácido ou nucleotídeo (formato FASTA)
Cria e Formata DB (dbstruct)
Busca similaridade pelo RAFTS3
Encaminha para o script RAFTS3groups com limiar de self-score
Agrupa as sequências com os self-scores superiores ao limiar
Monta a matriz com informação de Clusters totais (contall) e Clusters com mais de duas sequências agrupadas (contg2)
SAÍDA: (1) ou (2)

FIGURA 3. WORKFLOW BÁSICO DA FERRAMENTA RAFTS3GROUPS. Nota-se que ela pode ter duas saídas básicas: Dados gerados pelo algoritmo podem ser extraídos e analisados (1) ou os dados podem receber mais uma etapa de filtragem pelo *script dividecuster* e partir para análises de homologia (2).

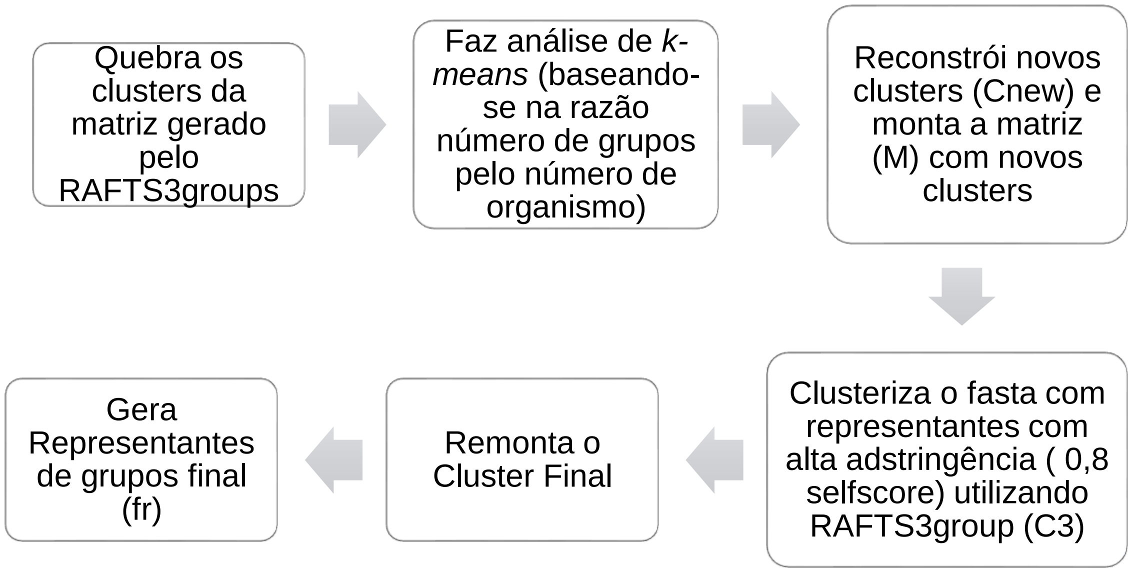

Quebra os clusters da matriz gerado pelo RAFTS3groups
Faz análise de k-means (baseando-se na razão número de grupos pelo número de organismo)
Reconstrói novos clusters (Cnew) e monta a matriz (M) com novos clusters
Clusteriza o fasta com representantes com alta adstringência ( 0,8 selfscore) utilizando RAFTS3group (C3)
Remonta o Cluster Final
Gera Representantes de grupos final (fr)

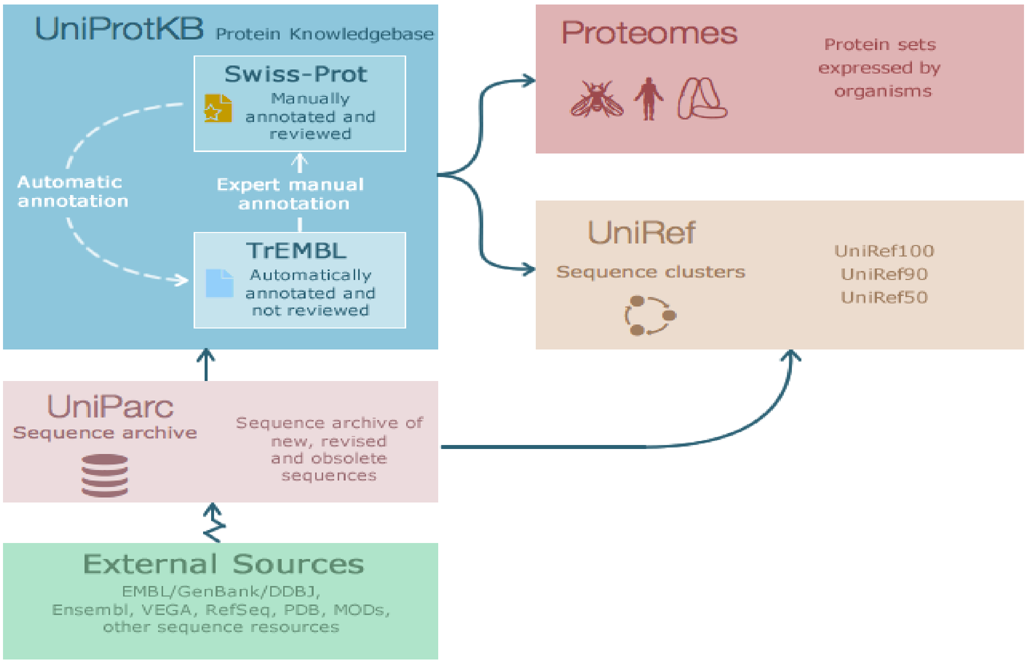
FIGURA 4. WORKFLOW BÁSICO DA ESTRUTURA DO ALGORITMO *DIVIDECLUSTER* PARA CLUSTERIZAÇÃO DE ORTÓLOGOS.

Após os clusters serem gerados pela ferramenta RAFTS3groups, o algoritmo *DivideCluster* reune informações adicionais das sequencias biológicas em que informações de organismos e de grupos funcionais são necessários para a clusterização voltada para análise de ortologia.

RAFTS3groups pode ser facilmente trabalhada sem a necessidade da segunda etapa, servindo como uma ferramenta de clusterização de homólogos, conforme descrito na primeira etapa. A próxima etapa utiliza, como aplicação complementar à ferramenta RAFTS3groups,  o *DivideCluster script* voltado para o agrupamento de grupos de ortólogos. Durante a segunda etapa, os *clusters* gerados pelo algoritmo RAFTS3groups em forma de matriz, juntamente com informações adicionais contidos no arquivo FASTA carregado, é *parseado* com informações de grupos e de organismos do conjunto inicial de formato padrão FASTA. Os *clusters* são refeitos, quando *número* de *clusters* são maiores que o *número* de organismo, portanto é feito uma re-clusterização com o auxilio do cálculo k-*means*. A matriz com as informações de *clusters* gerada por RAFTS3groups é reconstruída e é gerada uma nova matriz com os *clusters* novos ou reagrupados.

### 2.3.2 Universal Protein Resource (UniProt) e o programa Cluster Database at High Identity with Tolerance (CD-HIT)

O UniProt (Disponível em: www.UniProt.org/help/about) é uma colaboração entre o Instituto Europeu de Bioinformática (EMBL-EBI), o Instituto Suiço de Bioinformática (SIB) e o Recurso de Informações de Proteínas (PIR), onde mais de 100 pessoas estão envolvidas em tarefas distribuídos em curadoria, desenvolvimento de software e suporte (APWEILER, 2004). O UniProt é um recurso abrangente para dados de sequência e de anotação de proteínas. É composto por três principais bancos de dados: o UniProt Knowledgebase (UniProtKB), o UniProt Reference *Clusters* (UniRef) e o UniProt Archive (UniParc) conforme ilustrado na FIGURA 5. UniProt Knowledgebase (UniProtKB) é o ponto central para o recolhimento de informações funcionais sobre proteínas, com anotação exata, consistente e rica (HUANG, 2010). Além de capturados os dados de núcleo, obrigatórios para cada entrada de UniProtKB, o máximo de informações possíveis de anotação é adicionado ao Banco. Nele estão contidos dois Bancos principais, o TrEMBL, de curadoria automática, contendo mais de 64 milhões de sequências e o Swiss-Prot contendo mais de 550 mil sequências depositadas, um banco curado além de computacionalmente, também manualmente.



UniProtKB Protein Knowledgebase
Swiss-Prot
Manually annotated and reviewed
Automatic annotation
Expert manual annotation
TrEMBL
Automatically annotated and not reviewed
Proteomes
Protein sets expressed by organisms
UniRef
Sequence clusters
UniRef100
UniRef90
UniRef50
UniParc
Sequence archive
Sequence archive of new, revised and obsolete sequences
External Sources
EMBL/GenBank/DDBJ, Ensembl, VEGA, RefSeq, PDB, MODs, other sequence resources

FIGURA 5: COMPOSIÇÃO ESTRUTURAL DO BANCO UNIPROT

O algoritmo CD-HIT (disponível em http://weizhongli-lab.org/cd-hit/download.php) é um pacote de algoritmos próprio muito utilizado para clusterização de sequências de nucleotídeos e de proteínas. Segundo seus desenvolvedores, é uma ferramenta muito rápida e consegue lidar com bancos de dados extremamente grandes (LI & GODZIK, 2006). Também visa reduzir a redundância de informações e aumentar a velocidade na geração de *clusters*, pois é um método com várias funções de filtragem e de clusterização (LI & GODZIK, 2006). Sua importância é destacada na consolidação do Banco de dados UniProt, mais especificamente na consolidação do Banco UniRef (Abrange o Uniref 100, Uniref 90 e Uniref 50).

Uma limitação do CD-HIT é que seu filtro de palavras não pode ser usado abaixo de certos limiares de *clusters*. O Algoritmo do CD-HIT sorteia um input de tamanho longo para curto (no mínimo 11 resíduos de aminoácidos) e a primeira sequência é classificada e comparada como a primeira sequência representativa. A partir dela a classificação se baseia na similaridade (identidade global da sequência).

### 2.3.3 USEARCH e UCLUST

USEARCH (disponível em http://www.drive5.com/usearch/) é uma ferramenta utilizado por milhares de usuários pelo mundo e que combina vários algoritmos em um único pacote, como de busca de alto rendimento e de clusterização, sendo mais rápido que o algoritmo BLAST e cerca de 10 a 100 vezes que o algoritmo CD-HIT. (EDGAR, 2010). Possui versões livre e comercial, sendo que a livre é disponibilizada em 32-bits a todos os usuários.

Possui um algoritmo próprio, o UCLUST, que consiste em encontrar centroides (sequências representativas) e a partir delas são gerados grupos representativos definindo-se um limiar de identidade (T). O limiar de identidade pode ser visto como o raio de um *cluster* (FIGURA 6) em que sequências próximas são agrupadas. As identidades são calculadas usando um alinhamento global.

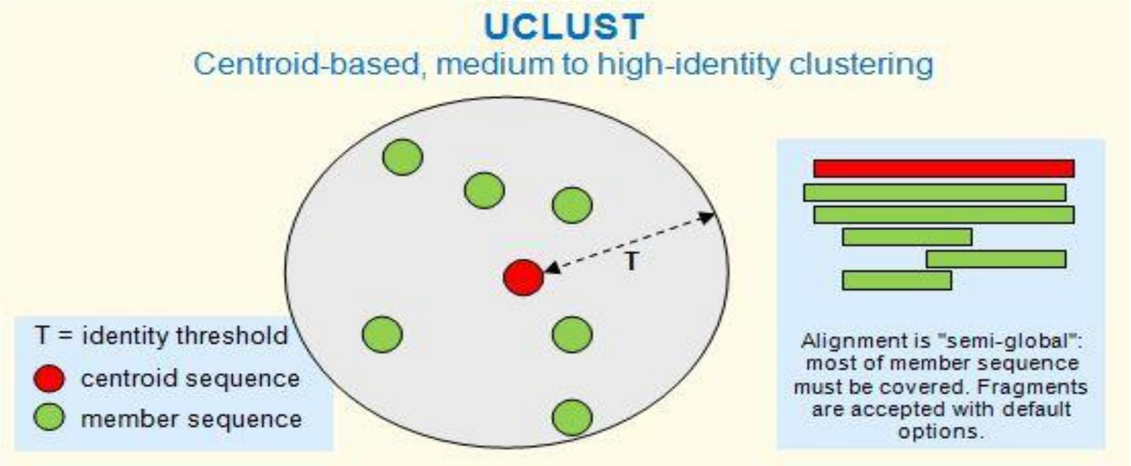

UCLUST
Centroid-based, medium to high-identity clustering
T
T = identity threshold
centroid sequence
member sequence
Alignment is "semi-global": most of member sequence must be covered. Fragments are accepted with default options.

FIGURA 6. ESQUEMATIZAÇÃO DO FUNCIONAMENTO BÁSICO DO ALGORITMO UCLUST

O algoritmo próprio do UCLUST seleciona centroides (círculo em vermelho), a partir delas sequências próximas são agrupadas (círculos em verde) e são gerados grupos representativos, respeitando-se um limiar de identidade (T) (EDGAR, 2010).

UCLUST é um algoritmo guloso (ou ganancioso), e a ordem de seqüências de entrada é importante. No comando *cluster_smallmem*, seqüências são processadas na ordem em que aparecem no arquivo de entrada. Se a próxima seqüência corresponde um centróide existente, é atribuído a esse *cluster*, caso contrário torna-se o centróide de um novo *cluster*. Significa que as seqüências devem ser ordenadas para que os mais apropriados centróides tendem a aparecer mais cedo no arquivo. UCLUST é eficaz em identidades superiores a 50% para proteínas e superiores a 75% em nucleotídeos. Em identidades inferiores, este tipo de método é questionável porque degrada qualidade de alinhamento e homologia não pode ser determinada de forma fiável de um alinhamento.

# 3 ARTIGOS CIENTÍFICOS

## 3.1. New Tools in Ortholog Analysis: A Brief Review of Promising Perspectives

### Abstract

Identifying homology relationships between sequences is essential for biological research. Within homology, the orthology analysis of sequences is of great importance for computational biology and annotation of genomes and the phylogenetic inference and is growing with the increase in new sequences that are deposited in databases. Because of this growth, since 2007 due to growing demand in the study with sequences deposited in biological databases, researchers began to analyse the profile of methodologies and of computational tools, in order to highlight the most promising ones for the prediction of orthologous groups.Through various searches in Google Scholar, PubMed and Expasy databanks, we have selected the latest techniques and tools that solve most problems in the detection of orthology, which is what motivates the present study, covering more than 100 articles. We listed the main computational tools, between 2011 and early 2017, and selected the 14 tools that differ in the type of orthology analysis, highlighting their key features and showing the biases that each seeks to resolves. Unfortunately, we observed that several tools are still using the default metodology BLAST ‘‘all-against-all’’ which bring some limitations, like the limitation of queries, computational costs and costs in terms of longer times needed for the analysis. However, there is a new approach in visualization tool as in the case of OrthoVenn, and the attempt of automation of work as proposed by the SPOCS and the ReMark tools or the attempt to minimize time and expense as in computational proteinOrtho method. These appear as viable alternative tools depending on the basic need of the user in the orthologs studies. We expect this review to assist and direct researchers in selecting the most appropriate tools for future scientific work, facilitating and elucidating their analyses involving orthology.

Key-words: Orthology, Comparative analysis, Genomic dynamics, Orthologous tools, Phylogeny.

## 3 Introduction

Identifying the homology relationship between sequences is essential for biological research [9]. Orthology analyses that consist, in finding out if a pair of homologous genes are orthologs – i.e.: resulting from a speciation - or paralogs - i.e.: resulting from a gene duplication - is very important in computational biology, genome annotation, and phylogenetic inference [20]. Because of this, the highlight of the present research was the development of computational tools that aim at facilitating this field of study.

The process of orthologs detection, besides being closely related to comparative analysis and genomic dynamism, is also an extremely important field of study for helping to improve the functional annotation of various organisms [12] and it is still very important to elucidate processes evolving the appearence of species [23]. An accurate orthologyassignment is a crucial step for comparative genomic studies [18] and then, in some cases, there is a need for tools that analize closely related species by pangenomas [10] or for the creation of tools that use different strategies like the post-translational modifications proteins (PTMs) for a better orthology inference[4].

Since the early studies involving the establishment of techniques for inferring orthology, the main difficulty was the lack of a methodology and of a tool to be fully reliable in assemblying orthologous sets of data. It was only in 2007, that the first study about the sensitivity, accuracy and performance methods in detecting these groups arose [1] thus consecrating methodological "gold standards" as adaptations of - among others the Markov Cluster Algoritm models of the Basic Alignment Search Tool (BLAST) algoritm **;** of Reciprocal Best Hits (RBH)**;** of Correlation Coefficient-based Clustering (COCO-CL), Correlation Coefficient-based Clustering (COCO-CL); of Automatic Clustering of Orthologs and In-paralogs (Inparanoid) leading to the appearance of various tools such as the a Markov Cluster algorithm for grouping proteins into multi-species of othologous groups (OrthoMCL). BLAST tool and adaptations were consolidated in subsequent studies [5,14] resulting in several publications and in the creation of large biological databases containing Ortholog *Clusters* such as the *Clusters* of Orthologs Groups/euKaryotic Orthologous Groups (COG/KOG), Ortholog Data Bank (OrthoDB) and eggNOG [15].

However, it is difficult to detect ortholog groups and there is no effective tool for detecting these groups, but rather a set of tools that meet certain computational demands and interests of its users [1]. Also perceived was the need of many improvements still to be made for a more accurate orthology prediction using these tools building or upgrading the orthologous relationships between genomes requires a lot of computational effort and a lot of time [19], besides, relating orthology between organisms having distant kinship origins, for instance, still remains a remarkable challenge [6].

All of this is gets worse when there is a need to include a large number of sequences to be analyzed in order to infer orthology [2]. Another problem is the application of a high-level of programming knowledge on the part of researchers to analyze data, which hinders the smoothness of the work flow. Some methodologies and tools, like the consolidated as BLAST all-vs-all, RBBH, OrthoMCL, demand a high computational cost [17] that will add weigh on the capabilities of normal hardware and will end up requiring access to the resources of supercomputers [16]. Another factor that directly influences the demand for better tools is the ever-increasing number of genomes that are deposited in large biological sequences in databanks and that can be compared simultaneously. This requires more efficient software tools [16, 7] because those such as BLAST and Inparanoid fail when orthology is involved, but the level of conservation among orthologs is low, and therefore this requires a sophisticated manual intervention and makes it difficult to automate the process [22]. Besides all that also comes the need to develop tools to improve the sensitivity in detecting orthologous groups [9].

Those are the most important needs and because of them several research groups are putting in great effort to develop new tools to improve and facilitate analysis involving orthology and may also contribute to advances in later studies. Therefore, the latest tools already available should gain prominence in the scientific field. Reviews of recent ortholog tools are gaining prominence, so much that, in 2015, came the first review tool involving homology pan genomes [21,24].

A number of free tools and web servers are available for pan genome analysis, but each of them suffers from one or the other limitations, leaving rooms for further improvement [3]. There has been, therefore, a pressing need for development of a new computational pipeline, which will not only offer fast and effcient forms for construction of the pan genome through clustering of orthologous gene families and but also enable various downstream analyses such as mapping of the core, accessory and other relevant analyses [3]. An option for applying the tools to a subset of the total dataset may facilitate identifcation of exclusive genetic features that can find similar groups like ortholog groups or discriminate between different serological, ecological or pathogenic groups [3].

In order to compile our review we focused on the most important and recent tools that have been developed with high expectatives for the study of orthologs, in order to bring the lastest advances in the development of more effective, fast and multi-tasking tools for the processing of homologous orthologous data sequences.

## 2 - Highlighting Main Tools and Methodologies

Due the importance and the growth in studies aimed at the development of new orthology techniques, since 2014 we have been monitoring various techniques of various techniques and computational tools for predicting protein or orthologous genes at the most different levels of study. After 2011 and up to the present day, we have observed a growing number of new computational tools emerging in publications (Figure 1).

The information derived from multiple banks of scientific publications (such as PubMed, Google Scholar and ExPASy) were filtered in a way that would allow us to find the lastest tools, softwares or recent packages that were aimed at resolving the issue of creation and consolidation of ortholog groups, but that would also solve problems left by older methodologies and tools. Each individual article was carefully examined analyzing contents of abstracts and scientific magazines, totaling more than 100 articles in order to compile the present review. Many, among the collected items, belong to BMC Bioinformatics, Oxford Journals of Bioinformatics Session, Evolutionary Bioinformatics and Genome Biology.

In this study, we found several papers highlighting tools that have proved promising for incorporating new methodologies or adapting already consolidated ones. Some even include specific algorithms developed in consideration of the fact that there are various problems to be solved in ascertaining orthology between organisms. Hereafter, we list 14 among the more recent major tools developed or in development highlighting their key features, described - in the following paragraphs and in Table 1 and Table 2, - algorithms involved, advantages and disadvantages and possible tool solutions in order to minimize specific problems encountered in the prediction of orthologs.

## 2.1 – morFeus: a program to detect remotely conserved orthologs using symmetrical best hits and orthology network scoring

The morFeus tool, published in 2014, presentes, as its key strategy, the search of remotely conserved orthologous groups. Morfeus selects sequences based on their alignment similarity of query using orthology tests based on research iterative reciprocal BLAST mode and calculates a network score to the resulting network orthologs which is a measure dependent on *e-value* implementation [22].

Given the variability between compared sequences and heuristics of BLAST, the *e-value* aims at providing users with the assurance that the score given to a particular *hit* did not occur randomly [13]. The performance of morFeus is comparable to other state-of-the-art orthology methods. Besides, some of its results have already been experimentally demonstrated by its developers that proved equivalent in organisms with comproved orthology thus fulfilling the criteria of the orthology-function conjecture [22]. This tool can be used both locally, and in this case only on Linux platforms, and via web -service.

## 2.2 - OrthAgogue: an agile tool for the rapid prediction of orthology relations in a large data set

One of the main problems concerning the most discussed in tools of orthologous clustering genes and proteins is the low computational performance and the high consumption of time that these tools need [8]. Considering this context, orthAgogue was developed and published in 2013. The tool works in multithreaded and is concerned with determining at high speed_relationships between sequences of genes or proteins of various species operating through a flexible and easy* command line interface. The best high-scoring pairs in BLAST output (HSP) is applicated in its algoritm to search orthologous groups [8].

In one of the papers that was considered to copile this review, orthAgogue is compared to the OrthoMCL tool, and, among other things, it point at computational limitations of the “gold standard” tool, such as the high consumption of RAM and processing when there is need to work with large volumes of data. Therefore orthAgogue is particularly convenient when working on large amounts of data with computers of limited capabilities. OrthAgogue is available for Linux platforms, being a tool developed in C+ language.

## 2.3 - OrthoInspector: comprehensive visual exploration in orthology and paralogy analysis

The OrthoInspector is a software system, published in 2011, which incorporates a unique algorithm for rapid detection of orthology and inparalogy between different species. First, the results of a BLAST *'all-versus-all'* is provided by the user and is parsed to find all the BLAST best hits for each protein and to create the groups of inparalogs. After that, the inparalog groups for each organism are compared in a pairwise fashion to define potential orthologs and/or in-paralogs. In the end, best hits that contradict the potential orthology between entities are detected [17]. In comparison with traditional methods, like orthoMCL and Inparanoid, the software shows improvements in the detection sensitivity with a minimal loss of specificity. [17]. Besides, the biggest difference of the package is that multiple visualization tools have been developed to facilitate analysis and study in depth based on its estimates, which allows for greater ease of consultation of the obtained data.

The OrthoInspector package, developed in Java, is compatible with any operating system, provided one has the JVM (Java Virtual Machine) preinstalled on the Operating System (OS). The tool is still in development, with a version 2.0 that has been available since 2014.

## 2.4 - OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy

OrthoFinder is an algorithm published in 2015, which aims to solve the bias accuracy in detecting orthologous groups. For this, it goes through several steps: (1) The unknown orthogroups that the algorithm must recover, shown as a gene tree. (2) The BLAST search of all genes against all genes. (3) Gene length and phylogenetic distance normalisation of BLAST bit scores to give the scores to be used for orthogroup inference. (4) Selection of putative cognate gene-pairs from normalised BLAST scores. For this algorithm is divided into several stages involving BLAST 'all-against-all' phylogenetic tree construction and use of MCL algorithm [9].

Using sets of real reference data demonstrated that OrthoFinder is more accurate than other methods of inference of orthologous groups already consolidated, such as OrthoMCL, TreeFam, eggNOG e OMA, between 8% e 33%. The methodology of this tool is based on the fact that the group contain all the genes descendants from a single gene in the last common ancestor of the species whose genes are being analyzed.

This setting prevents confusing shared ancestry with other criteria that are not equivalent, such as the functional conservation.

The tool started to be developed in 2003, was patented in 2015 ( US20150284796 ), but only in 2015 the paper relative to its contributions was submitted. It is a simple algorithm, light and easy to use in Linux environment.

### 2.5 - Ortholog-Finder: A Tool for Constructing an Ortholog Data Set

To obtain ortholog data sets for performing phylogenetic analysis by using all open-reading frame data of species, was developed Ortholog-Finder. Identifying genuine orthologs among distantly related species it is the main feature, focusing on 5 types of filtering genes to obtain through horizontal gene transfer (HGT) and out-paralogs to predict orthologs groups: (1) HGT filtering, (2) out-paralog filtering, (3) classification of tree data, (4) tree splitting, and (5) E-value changing. After HGT filtering, the inferred HGT sequences and non-HGT sequences are saved separately, and the data can be used for other analyses. The software does not support the maximum likelihood method or Bayes method because the calculation times required for choosing the optimal substitution model and constructing phylogenetic trees are extremely long [11].

Published in 2016, its downloadable to Linux/Unix plattforms (it was tested on Ubuntu 12.04 LTS and CentOS 6.5) and requires BLAST+, FastTree, MAFFT, Gblocks, BioPERL, EMBOSS, mcl, OrthoMCL and JAVA Runtime package to runs.

### 2.6 - Orthograph: a versatile tool for mapping coding nucleotide sequences to clusters of orthologous genes

Published in 2017, the Orthograph makes orthology prediction using a graph-based. Its pipeline applies the Reciprocal Best Hits (RBH) search strategy given that complete information of the organisms gene when the repertoire is available (e.g. RNAseq) [18]. Using profile hidden Markov models and maps nucleotide sequences to the globally best matching cluster of orthologous genes, thus enabling researchers to conveniently and reliably delineate orthologs and paralogs from transcriptomic and genomic sequence data [18]. Orthograph solves problems suffers from algorithmic issues that may cause problems in downstream analyses and is foccused in RNAseq analysis being a easy to use tool and flexible to users.

The software is written in PERL and its package runs locally an Unix/Linux systems (including OS X) but dependences are needed to run (BLAST + package, MySQL, HMMER, Perl, MAFFT, SWIPE).

## 2.7 – OrthoVenn: a web server for genome wide comparison and annotation of orthologous *clusters* across multiple species

Focusing on comparative genomics study, OrthoVenn implemented in Java, tries to illustrate, using the Venn diagram to create an overlap between the *clusters* of orthologous groups, and the function and evolution of proteins in various species. The first tool publication also took place in 2015. OrthoVenn is a Web-only tool" for viewing wide comparisons of orthologous groups of genomes with an interactive view of the Venn diagram and provides a summary of high-level functions for sets that overlap, or do not overlap, orthologous genes. It is a tool composed of several methods such as MCL, BLAST all-versus-all and besides, for the identification of hypothetical orthology, OrthoVenn uses the OrthAgogue tool for identifying orthology and inparalogy relations [20].

OrthoVenn is avaliable on web server and it allows personalized protein analysis from defined species on the part of the user. OrthoVenn also includes in-depth views of *clusters*, using various sequence analysis tools.

## 2.8 - PanOCT: automated clustering of orthologs using conserved gene neighborhood for pan-genomic analysis of bacterial strains and closely related species

The pan genoma analysis of prokaryotes species or strains closely related, is the main function of the Pan Genome Ortholog Clustering Tool (PanOCT). It is a specific tool to find groups in closely related species in prokaryotic strains. PanOCT uses conserved genes neighborhood information ito separate recently diverged homologs that standard methodologies fail to find [10]. For this, its unifies various types of methodologies in its flowchart, including protein BLAST (BLASTP) all-versus-all, RBH and BLAST Score Ratio (BSR) to detect orthologs groups [10]. There are results of comparison between PanOCT and three commonly orthologus-search tools in commonly used graphs (Inparanoid, OrthoMCL and Sybill) using bacterial strains data available to the public and among them, it turned out that a high relationship between the results obtained, about 86 %.

Published in 2012, the tool makes co-orthologous *clusters* preferable for this type of analysis. Written in PEARL language, PanOCT is avaliable in Linux OS and it is still in development. It is avaliable in 3.23 version (September 2015 data).

### 2.9 - PhosphOrtholog: a Web-based tool for cross-species mapping of orthologous protein post-translational modifications

According to homology studies, there is a growing need for tools that facilitate cross-species comparison of PTM data. This is particularly important because functionally important modification sites are more likely to be evolutionarily conserved [4]. In this context, the web tool PhosphOrtholog was developed. Through an unconventional approach, using proteomic data, PhosphOrtholog works with four major implementations for analyzing for analyzing data reference maps of orthologs. (QUAIS 4 IMPLEMENTAÇÔES?)

Published in 2015, this application is designed for mapping known and new orthologous PTM sites from experimental data obtained from different species in a large-scale PTM sites. Built on jQuery, Python and R this tool was incorporated and designed in HyperText Markup Language 5 (HTML5) and available exclusively via Web.

### 2.10 PorthoDom: Domain similarity based in orthology detection

In order to minimize the time and computational requirements in comparative analyzes between various sequences of proteins that are available, there is the porthoDom tool. The tool is based on the functional similarity domain of the protein content but their way of comparison is to bring two new measures of similarity between proteins: cosine similarity (COS) measure and a maximal weight matching score. A COS measure is implemented to compute the distance between two domain arrangements of any length [2]. The cosine measure is a similarity measure often used for high dimensional spaces [2]. The measures show that domain content similarities are able to correctly group proteins into their families. By using domains instead of amino acid sequences, the reduction of the search space decreases the computational complexity of an all-against-all sequence comparison.

The implementation of porthoDom is released using Python and C++ languages and is available under the GNU GPL licence 3 [2]. PorthoDom has a higher performance than proteinOrtho ortholog tool, being 40% faster.

## 2.11 - PorthoMCL: Parallel orthology prediction using MCL for the realm of massive genome availability

PorthoMCL is a Parallel orthology prediction using MCL for the realm of massive genome. It is similar to that of OrthoMCL, however, instead of depending on an external database server, the pipeline uses a sparse file structure for more efficient data storage and retrieval. Increase the number of genes using the 'All-Against-All' BLAST and MCL methodologies to scan orthology [19]. First, PorthoMCL conducts allagainst-all BLAST searches in parallel by performing "individual-against-all" BLAST searches for every genome independently. Second, it identifies the best betweengenomes BLAST hits for each two genomes "A" and "B" in parallel by scanning the "individual-against-all" BLAST results. Third, the algorithm finds reciprocal best hits between every two genomes and calculates the normalized score in parallel. This is the most computationally intensive step in the algorithm, specifically, for each parallel process, PorthoMCL loads at most two best-hit files at the same time to reduce the memory footprint, and every best-hit file is only loaded once to lower the I/O costs. Finally, PorthoMCL finds within genomes reciprocal best hits and normalizes the score with the average score of all the paralog pairs that have an orthologs in other genomes. These step are embarrassingly parallel computing problems and do not require shared memory, process coordination or data exchange platform as used in orthAgogue [19]. The output of these steps are eventually collated to construct a sequence similarity graph that is then cut by the MCL program to predict orthologous and paralogous gene groups.

Published in 2017, the program runs on Linux/Unix (OS X) and Windows systems and requires PERL, BLAST, Python, MCL dependences.

## 2.12 – ProteinOrtho: Detection of Co-orthologs in large-scale analysis

The main objectives of the developers of ProteinOrtho were to significantly reduce the amount of memoy required for orthology analysis of proteins, (being comparatively as good as OrthoMCL and Multi-Paranoid) and to deal easily with a large volume of data. Its implements a BLAST-based approach to determine sets of co-orthologous proteins or nucleic acid sequences that generalizes the reciprocal best alignment heuristic [16]. Published in 2011 and being improved, ProteinOrtho is a autonomy and a handling large bacterial datasets using distributed computing techniques when running on multi-core hardware. Its performance is comparable to that of OrthoMCL and, compared to OrthoMCL, due to its low computational request, ease of usage and good efficiency.

ProteinOrtho is one of the most cited in scientific papers (so far, it has over 40 citations) and it is avaliable in 5.11 version. It is easily used on Linux via terminal, but needs python, Perl and BLAST + pre-installed and requires a 64 bit OS.

### 2.13 - ReMark: an automatic program for clustering orthologs flexibly combining a Recursive and a Markov clustering algorithms

Identifying orthologs automatically is very useful for functional annotation, and studies on comparative and evolutionary genomics. The program ReMark is a fully automatic tool for clustering orthologs by combining a Recursive and a Markov clustering (MCL) algorithms [12]. Published in 2011, this tool is divided in two main steps: (1) The ReMark detects and recursively *clusters* ortholog pairs through reciprocal BLAST best hits between multiple genomes running soft-ware program (RecursiveClustering.java). (2) Then it employs MCL algorithm to compute the *clusters* (score matrices generated from the previous step) and refines the *clusters* by adjusting an inflation factor running software program (MarkovClustering.java) [12].

The program was developed in Java scripts, it works in cross-platform, since with the JVM pre installed on machine.

### 2.14 - SPOCS: software for predicting and visualizing orthology and paralogy relationships among genomes

Published in 2013, Species Paralogy and Orthology Clique Solver (SPOCS) implements a graph-based ortholog prediction method to generate a simple tab-delimited table of orthologs and in addition, HTML files that provide a visualization of the predicted ortholog/paralog relationships to which gene/protein [7]. SPOCS proceeds through three main stages: First, it executes a series of BLAST runs between every pair of species to identify reciprocal best hits, allowing subsequent SPOCS runs that include some of the same *n* species to avoid performing BLAST runs if they already exist. In the second stage, SPOCS uses the BLAST results to generate an orthology/paralogy relationship graph based on merging the pairwise ortholog and in-paralog relationships. Finally, SPOCS identifies cliques in each graph by breaking it into subgraphs and using the branch and bound clique-finding algorithm [7].

It is a flexible method for quickly and accurately predicting orthologs expression. Another plus point of the tool is that it can be worked via the web, and also locally in Linux or Mac OS X, but dependent on the boost C++ libraries and the previously installed BLAST.

## 3 – Discussion and Results

We made a critical analysis regarding use of each tool, collecting information from their presentations and their creators. We exposed their differentials and criticisms made about them about its main features, disavantages (restrictions) and usability.

MorFeus aims to get orthologs when there is difficulty in meeting orthology relationship between evolutionarily distant sequences. It runs on the Web, but only a sequence of input is run with ID mandatory the RefSeq. The configuration options are the choice of a particular kingdom (Archae, Bacteria, Fungi, Metazoa) or of the whole Bank, e-value (default is 100) and the output is sent to the email address of the registered user. Unfortunately, the tool is made available to run only locally, it has not been since March 2014 and it still has several dependencies to be rotated as python, biopython, networkx, gnuplot and BLAST + and the need to create a user.

OrthAgogue is a tool focused on the search of orthology prediction in a large set of data. It is available in 32 and 64-bit versions and it is currently at version 1.0.2 (updated in July, 2013). It depends on the library Intel TBB e library hash cmph. Thread number settings, threshold overlap are some of its differentials, which act on software agility it Is a relatively simple tool to use because it does not rely on very labourious prerequisites, however there is still the need for the input file to be a tabular file generated by the BLAST algoritm.

OrthoInspector is a tool that differentiates by offering a unique and fast algorithm of ortologia and in-paralogy. It is currently in version 2.21 (updated August 2015). The needed prerequisites to run it are, in addition to the settings of the algorithm, such as the case of input in XML format, the need of the BLASTP+ package, and the creation of a Database in Postgresql or MySQL, and the need of the JAVA package pre-installed makes its handling more difficult.

OrthoFinder was developed to solve fundamental biases in whole genome comparisons, improving the accuracy of the inference of orthologs groups. It works as a single command that takes as input a directory of FASTA files (one per species) and, with the help of statistical algorithms, it generates output files containing genes of orthologs groups of these species. This mechanism is interesting because it seeks to minimize the bias of the length, previously undetected gene in orthogroup (orthologs group) inference, resulting in significant improvements in accuracy. It is at version 0.7.1 (updated on July 2016) and dependent on packages Python, BLAST+, MCL graph clustering algorithm, MAFFT and FastTree previously instaled.

Ortholog-Finder is a program developed to constructing ortholog data sets for phylogenetic analysis. Results from its developers suggests can tolerate gene loss after gene duplication and HGT events, because most of the phylogenetic trees were accurately reproduced even when these events occurred. It was wrote in PERL and it is compatible with Linux/Unix platfforms and needs BLAST, ClustalW, MAFF and BioPERL dependences to runs. The program does not support the maximum likelihood method or Bayes method.

Orthograph, with its specific algorithm, it solves this issue that earlier implementations of graph-based BRH mapping strategies suffered from, while maintaining the high sensitivity and accuracy of the BRH approach. Unfortunately, needs a BLAST algoritms and lot of others dependences.

OrthoVenn, using the interactive Venn diagram in the generated clusters views, is a tool that seeks orthology between multiple sequences between different species. One can select up to six user-species and analyze them against the DataSet. It brings Gene Ontology information with each protein function, relating to the respective clusters generated. The pipeline integrates various methodologies for the inference of groups like BLAST "all-vs-all", MCL and even a predictor of hypothetical proteins, which makes this tool somewhat time-consuming and it implies input limit for its implementation via Web server.

PanOCT tool, developed in PERL, stands out for not using traditional methods in graph-based detection of orthologs and because it is considered a high output management tool. It uses conserved gene neighborhood (CGN) strategy to improve accuracy in the clusters generated by the algorithm. It is a tool for pan-genomic analysis of prokaryotic species or closely related strains, therefore, it presents difficulty when organisms are too far apart. It depends on the PERL packages, BLAST + and it is limited to an analysis of up to 25 genomes (if the machine has a setting of 14 GB of RAM). It closely resembles the clustering tool with several interesting execution options such as *e-value* threshold, cut-off identity, creation of files with paralogs groups, BLAST standardization score histogram, window size on either side of match to use CGN, among others. We believe that the various options are necessary due to the preference of the Group tool just the co-orthologs with the same genomic context, and additional information should be reported indicating the co-ortologos relationship. Currently being made available in version 3.23 (updated on july 2016).

Phosphortholog is another tool available exclusively on the Web for mapping orthologous protein species from post-translational modifications (PTMs). To this end, it has as reference the database UniProt/Swiss-Prot, where information about proteins is collected and its algorithm uses score based on the BLOSUM62 matrix for alignments between sequences. One of the difficulties, unfortunately, is that the input is in comma separated file format and is restricted to just the proteomas of humans, mice, rats, flies.

ProteinOrtho significantly reduces the amount of memory needed for orthology analysis in comparison with existing tools (OrthoMCL and Multi-Paranoid). It finds co-orthologs on big banks containing different species, specifically designed to handle hundreds of species together containing milions of proteins. However, unfortunately, it still depends on libraries BLAST +, PERL and Python to run. It is available in version 5.1 (April 2016).

PorthoDom is composed of two parts, a C program computing the pairwise domain similarity and a python gluing everything together. PorthoDom is a python wrapper using protein domain to speed up proteinOrtho. It performs the domain annotation of their protein sequences, or one can use the existing annotation in Pfam format. The clusters are used as orthologs subspace search candidates, that is, sequences of proteins with similar domain arrangement are grouped by species and proteinortho is performed in these sub-groups. The different results of the proteinOrtho runs are gathered and a default output file proteinOrtho is created. Therefore the tool is a bit laborious to be performed, and there is a need for pre-installed python packages, Pfam database and HMMER package, in addition to the ProteinOrtho tool.

Note that it is (its) a tool for groups flexibly orthologs through a parameter adjustment according to the user's interest and it makes the process more automated. It combines Recursive and Markov clustering (MCL) algorithms and it uses the Reciprocal BLAST Best Hits (RBBH) model between multiple genomes running *RecursiveClustering.java* software on the first step. Therefore it employs the MCL algorithm to calculate the clusters (scoring matrices generated from the previous step) and it refines the adjusting inflation factor by running *MarkovCLustering.java* software. The tool has not been updated since march 2011 and it also features JAVA dependencies, Ant (JAVA) library and previously installed BLAST.

PorthoMCL is a fast tool with low requirements for identifying orthologs and paralogs in any number of genomes. Its uses the same mathematical basis as OrthoMCL to investigate orthology among genomes, it is much faster and a more scalable tool when handling a very large number of genomes. PorthoMCL can facilitate comparative genomics analysis through exploiting the exponentially increasing number of sequenced genomes. Although fast and easy tool, requires BLAST, PERL and Python package.

SPOCS, among all the tools that were presented, emerges as an alternative to automate the process of orthologs detection, without the need for multiple steps. It also considers the ability of the hardware, requiring a minimum of 8 GB RAM and Quadcore processing in 64-bit based systems. The software will take a set of protein FASTA files (one per species genome), and an optional additional FASTA to serve as an outgroup (a species that should be more distantly related to the species of interest than any of the species of interest are to each other). BLAST is required to generate the reciprocal best hit results for every pair of species. SPOCS then merges these results identifying orthologs using the graph-based concept of cliques. SPOCS needs boost C++ and BLAST previosly installed and it generates text or HTML outputs.

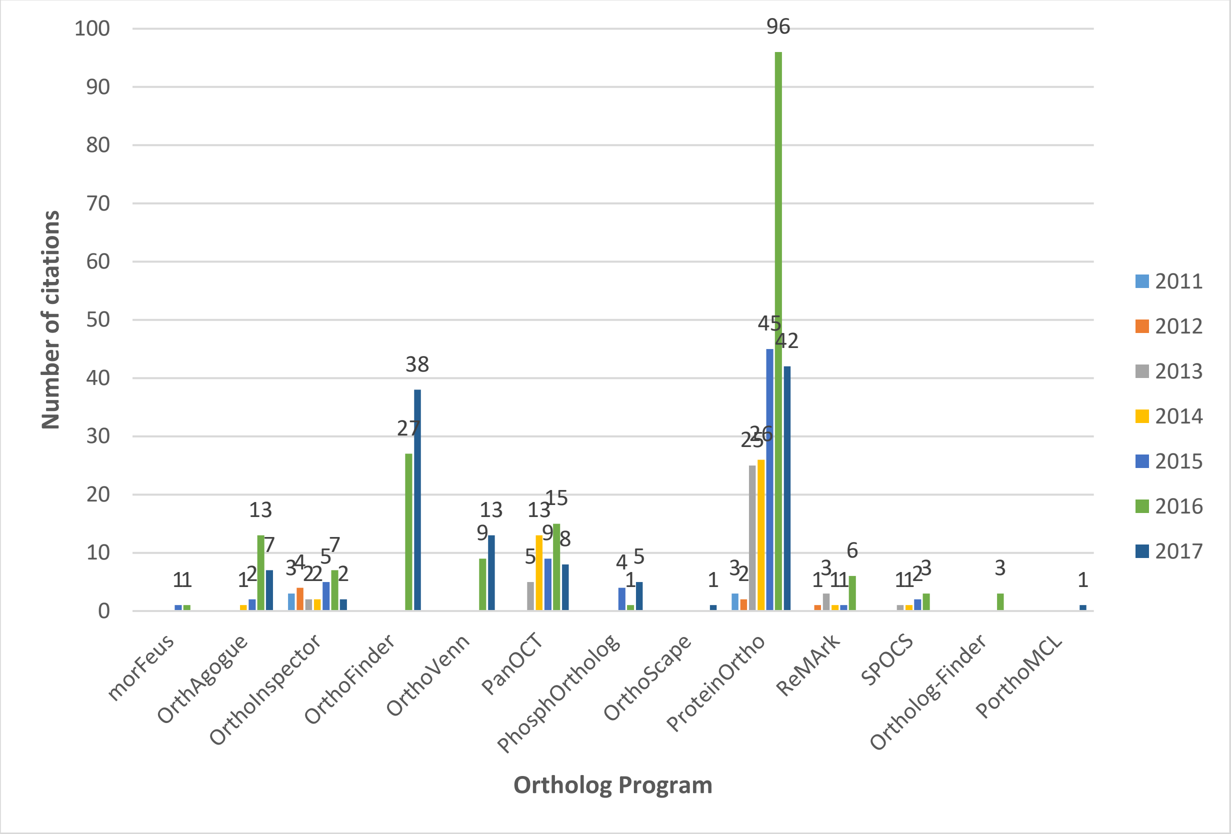


Figure 1 - Growth of Number of the citations by orthologs tools from 2011 to 2017. A brief relationship between the number of citations per year for each tool. It is observed that some of the tools have citations more than others in other, such as, ProteinOrtho, OrthoInspector, OrthoFinder and PanOCT revealing a good acceptance of tools in other works across the years.

Table 1 – Software tools features for Orthologous studies since 2010 at 2016

| Tool | Main Features | Platform | Implementation | Disponibility | Ref. |
|---|---|---|---|---|---|
| **MorFeus** | Calculates a network score to the resulting network orthologs to find remotely stored orthologs proteins | Linux/Unix; web-server: Platform-independent | python, biopython, networkx, gnuplot and BLAST+ | http://bio.biochem.mpg.de/morfeus/; (web)<br>https://sourceforge.net/p/morfeus/ (installer) | [22] |
| **OrthAgogue** | High speed of the homology relationships in large data sets | Linux/Unix | BLAST, cmph and TBB | https://code.google.com/p/orthagogue/. | [8] |
| **OrthoInspector** | Incorporates a unique algorithm for rapid detection of orthology and in-paralogy | Cross-platform (Java) | BLASTP+ package, JAVA, MySQL | http://www.lbgi.fr/orthoinspector/ | [17] |
| **OrthoFinder** | Solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy | Linux/Unix | BLASTP+ package, python, MCL graph clustering algoritm, MAFF and Fastree | http://www.stevekellylab.com/software/orthofinder | [9] |
| **Ortholog-Finder** | Identifying genuine orthologs among distantly related species performing phylogenetic analysis by using open-reading frame data | Linux/Unix | BLAST+, MAFFT, BioPERL, OrthoMCL, JAVA, ClustalW | http://www.grl.shizuoka.ac.jp/~thoriike/Ortholog-Finder | [11] |

| | | | | | |
|---|---|---|---|---|---|
| **Orthograph** | It solves this issue that earlier implementations of graph-based BRH mapping strategies suffered from with its specific algorithm, while maintaining the high sensitivity and accuracy of the BRH approach Developed for a wide range of comparative genomic and transcriptomic analyses | Linux/ Mac OS X | BLAST +, PERL, MySQL, MAFF, SWIPE | https://mptrsen.github.io/Orthograph/ | [18] |
| **OrthoVenn** | Wide comparison and annotation of orthologous *clusters* across multiple species | Web-server | BLAST and MCL algoritms | http://aegilops.wheat.ucdavis.edu/OrthoVenn/ | [23] |
| **PanOCT** | Automated clustering of orthologs for pan-genomic analysis of bacterial strains and closely related species | Linux/Unix | BLAST+, PERL | https://sourceforge.net/projects/panoct/ | [10] |
| **PhosphOrtholog** | For cross-species mapping of orthologous protein post-translational modifications | Web-Server | BLOSUM62, comma separated file format (.csv) input | http://www.phosphortholog.com/ | [4] |
| **porthoDom** | To Speed up the detection of orthologs protein using domain sequences | Cross-platform | C program, Python, pFam Database, HMMER | http://www.bornberglab.org/pages/porthoda. | [2] |
| **PorthoMCL** | Designed for find orthologs in a large number of genomes | Linux and Unix (OS X) | BLAST, PERL, Python, MCL | http://ehsun.me/go/porthomcl/ | [19] |

| | | | | | |
|---|---|---|---|---|---|
| **ProteinOrtho** | Dealing with hundreds of bacterial species in set containing millions of proteins using low computer memory | Linux/Unix 64bits | BLAST+, PERL and Python | http://www.bioinf.uni-leipzig.de/Software/proteinortho/ | [16] |
| **ReMark** | Identify orthologs automatically by a parameter adjustment according to the user's interest | Cross-platform (Java) | Recursive and a Markov clustering (MCL) algoritms, Reciprocal BLAST Best Hits (RBBH), JAVA | http://dasan.sejong.ac.kr/~wikim/notice.html | [12] |
| **SPOCS** | Orthologs prediction method based on graph to generate a table can provide a visualization of the relationships between the ortologos | Web-Server; .inux/Mac OS X | BLAST+, C++ libraries | http://cbb.pnnl.gov/portal/tools/spocs.html (web)<br>http://cbb.pnnl.gov/portal/software/spocs.html (installer) | [7] |

TABLE 2 – Software tools usabilities

| Tool | Advantages | Disadvantages |
|---|---|---|
| **Morfeus** | Uses symmetrical best hits and orthology network scoring to detect remotely conserved orthologs. | A lot of dependences, its not updated since 2014. |
| **OrthAgogue** | A multithreaded C application for high-speed estimation of homology relations in massive datasets. | Needs the input file be a tabular file generated by the BLAST algoritm, needs a lot of dependences. |
| **OrthoInspector** | Incorporates an original algorithm, facilitate data query, and process automation. | Creation of a Database in Postgresql or MySQL, BLAST dependences. |
| **OrthoFinder** | Works as a single command that takes, as input a multiFASTA files (one per species). Minimize the bias of the length, previously undetected gene in orthogroup. | Needs a lot of dependences to run. |
| **Ortholog-Finder** | A program for constructing ortholog data sets for phylogenetic analysis. | A lot of dependences and the program does not support the maximum likelihood method or Bayes method. |
| **OrthoVenn** | Visualisation: Using the interactive Venn diagram in the generated clusters views. Brings Gene Ontology information with each protein functions. | Only web-server, limitation of queries. |
| **Orthograph** | Orthograph is easy to install and use and thereby facilitates comparative analyses of transcriptomic | Needs a lot of dependences to run like BLAST, MySQL, MAFFT, HMMer, SWIPPE to runs. |

|  |  |  |
|---|---|---|
|  | and other coding sequence for a wide range of comparative genomic and transcriptomic analyses. |  |
| **PanOCT** | Procariotic uses, Orthologs and co-orthologs relationships. | Depends on the PERL packages, BLAST+ and is limited to an analysis of up to 25 genomes. |
| **PhosphOrtholog** | Mapping between orthologous protein species from post-translational modifications (PTMs). Uses UniProt/Swiss-Prot DB reference. | Exclusively on the Web, needs comma separated file format and be restricted to just the proteomas of human, mouse, rat, fly. |
| **porthoDom** | Uses protein domain to speed up proteinOrtho. Uses Pfam anotation to accuracity. | Its a bit laborious to be performed, needs a lot packages, Pfam database and HMMER package, in addition to the ProteinOrtho tool. |
| **PorthoMCL** | Capability by identifying orthologs in a very large number of genomes and easy to use. | Although fast and easy tool, requires BLAST, PERL and Python package. |
| **ProteinOrtho** | Reduces the amount of memory needed to create orthologs groups, finds co-orthologs on big banks containing different species. | Depends on libraries BLAST +, PERL and Python to run. |
| **ReMark** | Makes the process more automated, using adjustment according to the user's interest. | The tool is not updated since march 2011. Needs BLAST and JAVA dependences. |
| **SPOCS** | Flexibility and automates the process of orthologs detection, without the need for multiple steps. | Needs boost C++ and BLAST previosly instaled and generates text or HTML outputs. |

## 4 - Conclusion

We provide an overview of the new methodologies and tools of analysis of sequences in the study of orthologs sequences and we hope to see improvements of the original software tools in future work by their developers in order to assist and direct researchers in selecting the most appropriate tool for their work, making available to the scientific world more reliable and faster results thus contributing to new works and criticism.

Turning to the analysis of the tools in the prediction of orthology, it was possible we noticed some peculiar features of each one and also, unfortunately, some problems still present and that should be solved even in latest methodologies, for instance, most tools still use the pipeline BLAST, which demand greater processing and therefore it restricts the number of sequences that can be analized. This is still present in new methodologies that claim to be more efficient than OrthoMCL or Inparanoid, for example. However, some of the new tools are promising in the study of homology, for instance ProteinOrtho, OrthoInspector, OrthoFinder and PanOCT which aims to optimize some specific limitation in orthology analysis, or a new way of viewing, using Venn diagram as in OrthoVenn, the attempt to automation of work as proposed by the SPOCS and ReMark tools, appearing as viable alternative tools depending on the basic needs of their user in orthologs studies.

## 5 - Competing Interesses

The autors declared that are no competing interesses with this review.

## 6 – References

[1] Altenhoff, A. M., & Dessimoz, C. (2009). Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods. *PLoS Computational Biology*, *5*(1), e1000262. http://doi.org/10.1371/journal.pcbi.1000262

[2] Bitard-Feildel, T., Kemena, C., Greenwood, J. M., & Bornberg-Bauer, E. (2015). Domain similarity based orthology detection. *BMC Bioinformatics*, *16*(1), 154. http://doi.org/10.1186/s12859-015-0570-8

[3] Chaudhari, N. M., Gupta, V. K., & Dutta, C. (2016). BPGA – an ultra-fast pan genome analysis pipeline. *Nature Publishing Group*, (April), 1–10. http://doi.org/10.1038/srep24373

[4] Chaudhuri, R., Sadrieh, A., Hoffman, N. J., Parker, B. L., Humphrey, S. J., Stöckli, J., … Yang, J. Y. H. (2015). PhosphOrtholog: a web-based tool for cross-species mapping of orthologous protein post-translational modifications. *BMC Genomics*, *16*(1), 617. http://doi.org/10.1186/s12864-015-1820-x

[5] Chen, F., Mackey, A. J., Vermunt, J. K., & Roos, D. S. (2007). Assessing Performance of Orthology Detection Strategies Applied to Eukaryotic Genomes. *PLoS ONE*, *2*(4), e383. http://doi.org/10.1371/journal.pone.0000383

[6] Chen, T., Wu, T. H., Ng, W. V, & Lin, W. (2010). DODO: an efficient orthologous genes assignment tool based on domain architectures. Domain based ortholog detection. *BMC Bioinformatics*, *11 Suppl 7*(Suppl 7), S6. http://doi.org/10.1186/1471-2105-11-S7-S6

[7] Curtis, D. S., Phillips, A. R., Callister, S. J., Conlan, S., & McCue, L. A. (2013). SPOCS: software for predicting and visualizing orthology/paralogy relationships among genomes. *Bioinformatics*, *29*(20), 2641–2642. http://doi.org/10.1093/bioinformatics/btt454

[8] Ekseth, O. K., Kuiper, M., & Mironov, V. (2014). OrthAgogue: an agile tool for the rapid prediction of orthology relations. *Bioinformatics*, *30*(5), 734–736. http://doi.org/10.1093/bioinformatics/btt582

[9] Emms, D. M., & Kelly, S. (2015). OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biology*, *16*(1), 157. http://doi.org/10.1186/s13059-015-0721-2

[10] Fouts, D. E., Brinkac, L., Beck, E., Inman, J., & Sutton, G. (2012). PanOCT: automated clustering of orthologs using conserved gene neighborhood for pan-

genomic analysis of bacterial strains and closely related species. *Nucleic Acids Research*, *40*(22), e172–e172. http://doi.org/10.1093/nar/gks757

[11] Horiike, T., Minai, R., Miyata, D., Nakamura, Y., & Tateno, Y. (2016). Ortholog-finder: A tool for constructing an ortholog data set. Genome Biology and Evolution, 8(2), 446–457. http://doi.org/10.1093/gbe/evw005

[12] Kim, K., Kim, W., & Kim, S. (2011). ReMark: An automatic program for clustering orthologs flexibly combining a Recursive and a Markov clustering algorithms. *Bioinformatics*, *27*(12), 1731–1733. http://doi.org/10.1093/bioinformatics/btr259

[13] Korf I., M. Yandell, J. Bedell; An Essential Guide to the Basic Local Alignment Search Tool; O' Reilly & Associates, Inc., Sebastopol, U.S.A.,2003;

[14] Kristensen, D. M., Wolf, Y. I., Mushegian, A. R., & Koonin, E. V. (2011). Computational methods for Gene Orthology inference. *Briefings in Bioinformatics*, *12*(5), 379–391. http://doi.org/10.1093/bib/bbr030

[15] Kuzniar, A., van Ham, R. C. H. J., Pongor, S., & Leunissen, J. A. M. (2008). The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics*, *24*(11), 539–551. http://doi.org/10.1016/j.tig.2008.08.009

[16] Lechner, M., Findeiß, S., Steiner, L., Marz, M., Stadler, P. F., & Prohaska, S. J. (2011). Proteinortho: Detection of (Co-)orthologs in large-scale analysis. *BMC Bioinformatics*, *12*(1), 124. http://doi.org/10.1186/1471-2105-12-124

[17] Linard, B. (2011). OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, *12*(11), 1471. http://doi.org/10.1186/1471-2105-12-11

[18] Petersen, M., Meusemann, K., Donath, A., Dowling, D., Liu, S., Peters, R. S., … Niehuis, O. (2017). Orthograph: a versatile tool for mapping coding nucleotide sequences to clusters of orthologous genes. BMC Bioinformatics, 18(1), 111. http://doi.org/10.1186/s12859-017-1529-8

[19] Tabari, E., & Su, Z. (2017). PorthoMCL: Parallel orthology prediction using MCL for the realm of massive genome availability. Big Data Analytics, 2(1), 4. http://doi.org/10.1186/s41044-016-0019-8

[20] Ullah I., Sjöstrand J., Andersson P., Sennblad B.(2015). Integrating Sequence Evolution into Probabilistic Orthology Analysis. *DigitalaVertenskapliga Arkivet* urn:nbn:se:kth:diva-168167

[21] Vernikos, G., Medini, D., Riley, D. R., & Tettelin, H. (2015). Ten years of pan-genome analyses. *Current Opinion in Microbiology*, *23*, 148–154. http://doi.org/10.1016/j.mib.2014.11.016

[22] Wagner, I., Volkmer, M., Sharan, M., Villaveces, J. M., Oswald, F., Surendranath, V., & Habermann, B. H. (2014). MorFeus: a web-based program to detect remotely conserved orthologs using symmetrical best hits and orthology network scoring. *BMC Bioinformatics*, *15*(1), 263. http://doi.org/10.1186/1471-2105-15-263

[23] Wang, Y., Coleman-Derr, D., Chen, G., & Gu, Y. Q. (2015). OrthoVenn: a web server for genome wide comparison and annotation of orthologous *clusters* across multiple species. *Nucleic Acids Research*, *43*(W1), W78–W84. http://doi.org/10.1093/nar/gkv487

[24] Xiao, J., Zhang, Z., Wu, J., & Yu, J. (2015). A Brief Review of Software Tools for Pangenomics. *Genomics, Proteomics & Bioinformatics*, *13*(1), 73–76. http://doi.org/10.1016/j.gpb.2015.01.007

## 3.2 Rapid Alignment Free Tool for Sequence Similarity Search of Groups (RAFTS3groups) – A fast clustering software for large number of data and consistent ortholog protein detector

### Abstract

One of the main challenges involving biological sequences, essential and complex to bioinformatic research, is the analysis called of homology. The need to develop computational tools and techniques that can predict more efficiently ortholog groups and handle large volume of biological information is still a great problem in Bioinformatic studies. We don't have a single powerful tool in detecting groups still requiring a lot of effort and computing time. Tools already consolidated, as the BLAST 'all-against-all', RBH, OrthoMCL, demand a high computational cost and have low efficiency when used for orthogy, as they require manual intervention. We present a new approach for the analysis of homology, the RAFTS3 groups tool. We use as the database UniProtKB/Swiss-Prot with the clustering tools the UCLUST and the CD-HIT. RAFTS3groups tool proved to be more than 4 times faster than CD-HIT and comparable in volume to clusters and time with UCLUST tool. In Homology analysis we introduced a new clustering strategy for orthology, the *DivideCluster* algorithm built into the RAFTS3groups. Compared with the BLAST 'all-against-all', analyzing 9 complete genomes from *Herbaspirillum* spp. available on NCBI. RAFTS3groups was shown to be as efficient as a method, showing approximately 96% of the correlation between the results.

**Keywords:** Orthology, Comparative analysis, Clustering software, orthologous tools, RAFTS3groups.

## 1. Introduction

Since the emergence of large-scale genomic sequencing, in 2002, the analyses of genomes and proteomas began to be used and gained strength, mainly in recent years (Emms et al, 2015).

However, it was noted that, from that time, there was an exponential increase of more and more sequences to be deposited resulting in the necessity of creating large databases to store such information what we call Big Data (Emms et al, 2015). Such growth was responsible for bringing a major bottleneck to the field of analysis of sequences: the need to develop computational tools and techniques that can handle large volume of biological information. One of the main challenges involving biological sequences, essential and complex, is the analysis of homology.

Homology is the relationship between two organisms usually descending from, a common ancestor (Fitch, 2000).

As in anatomical structures, homology between DNA or protein sequences of different organisms, can share a common ancestry. Two segments of DNA may have shared ancestry because of a speciation event (orthologs) or an event (paralogs) (Koonin, 2005). Homology between proteins or DNA is usually inferred from the similatires of sequences.

Significant similarity is strong evidence that two sequences are related by divergent evolution from a common ancestor (Koonin, 2005). Multiple sequence alignments are used to indicate which regions of each string are homologous. The determination of orthogy or paralogy is related to events of Gene evolution (Fitch, 2000).

The study of homology is also a great field of computational process uses, statistical detection of logical or, in orthologs analysis. Closely relationships were related to comparative analysis between genomes and genomic dynamism between different organisms (Kim et al., 2011). The method still has been featured in analyses that help to elucidating evolutionary processes along with the emergence of species (Wang et al, 2015). Only in 2007, came the first study about the sensitivity, accuracy and performance of orthologs detection methodologies (Althenhoff & Dessimoz, 2009) and, with it, consecrating methodologies "gold standards". “Gold standards” are a pattern that serves as a comparison for other tests, for the purpose of assessing the accuracy of the same, in results that ensure maximum hits in order to establish the real diagnosis in orthologs groups consolidation (Dalquen, 2013), as adjustments of the Markov cluster models algorithm, adaptations of the algorithm Basic Local Alignment Search Tool (BLAST), Reciprocal Best Hits (RBH), Correlation Coefficient-based Clustering (COCO-CL), Automatic Clustering of Orthologs and In-paralogs (Inparanoid). This led to the emergence of several tools, such as the pipeline integrates a Markov Cluster algorithm for grouping proteins into multi-species orthologous groups (OrthoMCL), that were being consolidated in subsequent studies (Kristensen et al., 2001). The results exposed in various other publications and in the creation of large biological databases containing Ortologous *clusters*, as *Clusters* of Orthologous Groups/euKaryotic Orthologous Groups (COG/KOG), Ortholog Data Bank (OrthoDB) and the evolutionary genealogy of genes: Non-supervised Orthologous Groups (eggNOG) (Kuzniar et al, 2008).

Also detected was the need for many improvements still to be made for a more efficient orthologs prediction by these tools. Building or updating relationships between genomes of orthologs requires a lot of effort and time. In addition to that, very often, a relation between organisms of orthogy distant relationships, for example, continues to be a challenge (Chen & Wu, 2010). The work is compounded when, in the study, it includes a large number of sequences to be scanned for orthogy inference (Bittard-Feidel et al., 2015). Another problem is the application of a high level of programming knowledge on the part of researchers to analyze large volumes of data, which hinders the fluidity of the researches (Lechner et al, 2011). Some methodologies and tools, already consolidated, as the BLAST ‘all-against-all’, RBH, OrthoMCL, require a high computational cost (Linard, 2011) that goes beyond common capabilities of hardware and require access to resources of supercomputers (Lechner et al., 2011). Another factor that influences directly the demand for better tools is the ever-increasing number of genomes that are deposited in large banks of biological sequences which can be compared at the same time (Lechner et al., 2011). This requires more efficient software tools (Curtis et al., 2013), fact that tools such as BLAST and Inparanoid, fail when there is orthogy, but the conservation level is low, requiring manual intervention and that ends up making the process automation (Wagner et al., 2014). Before all that, there is the need of the development of tools to improve the sensitivity in the detection of orthologs groups (Emms & Kelly, 2015).

Due to the importance and key issue of the analysis of new methodologies and tools in the study of orthologs groups, it is a new alternative proposal of free tool of clustering, the RAFTS3group, to identify and consolidate groups of orthologous easily for the researcher, requiring less processing, consequently being faster, even dealing with large volumes of biological data.

## 2. Materials and Methods

### 2.1 UniProt e CD-HIT

The UniProt is a collaboration between the European Bioinformatics Institute (EMBL-EBI), the Swiss Institute of Bioinformatics (SIB), and the Protein Information Resource (PIR), It consists of three large banks of biological information, the UniProt Knowledgebase (UniProtKB) the UniProt Reference *Clusters* (UniRef), and the UniProt Archive (UniParc). Within this consortium, the UniProt Knowledgebase (UniProtKB) is the central point for gathering functional information about proteins, with accurate, consistent, and rich annotation. In addition to the required core data identified for every UniProtKB entry, the maximum of possible information of annotation is added to the Bank. The UniProtKB contained two major banks, the Translated EMBL Nucleotide Sequence Data Library (TrEMBL) database manually, only cured computationally, containing more than 64,000,000 of sequences and Swiss-Prot containing 550,299 sequences deposited (released at 02/15/2016), being a bank handled besides computationally, also manually. This data bank released was used in various selfscore analysis of RAFTS3group and in the comparison between the tools UCLUST and CDHIT.

The CD-HIT (available in http://weizhongli-lab.org/cd-hit/download.php) is a package of very specific algorithms used for the clustering of nucleotide and protein sequences (LI, W., & Godzik, 2006). It also aims to reduce the redundancy of information and increase speed in the generation of clusters, as a method with multiple filtering functions and clustering. Its importance is highlighted in the UniProt database consolidation, more specifically in the Bank consolidation UniRef (covers the Uniref 100, 90 and 50 Uniref Uniref).

The algorithm of CD-HIT draws a long size input for short (at least 11 aminoacids residues) and the first string is sorted and compared as the first sequence. From it the classification is based on the similarity (global sequence identity). It is a tool for high performance in high numbers of data of proteomes and one of the main tools that handle large volume of data, then was used in comparative analysis with RAFTS3groups and UCLUST against the UniProt/Swiss-Prot against 550,299 sequences to measure speed and clusterizing performance.

### 2.2 USEARCH e UCLUST

USEARCH (available at http://www.drive5.com/usearch/) is a tool used by thousands of users around the world that combines multiple algorithms in a single package, as high-performance search and clustering, being faster than the BLAST Algorithm (about 10 to 100 times) and the CD-HIT (Edgar, 2010). It exists in free and commercial versions and the free one is available, in 32-bits to all users. In particular, it has an algorithm for clustering of sequences, the UCLUST, which relies on finding centroides (representative sequences) and from them are generated representative groups by setting a threshold of identity. The threshold of identity can be seen as a cluster where sequences are grouped. The identities are calculated using a global alignment. UCLUST is a “greedy” algorithm, and the order of input sequences is important. In the *cluster_smallmem*, command strings are processed in the order in which they appear in the input file. If the next sequence matches an existing centroid, it is assigned to this cluster, otherwise it becomes the centroid of a new cluster. This means that the strings should be sorted so that the most appropriate centroids tend to appear earlier in the file. UCLUST is effective in identities over 50% for proteins and over 75% in nucleotides (Edgar, 2010). In lower identities, this kind of method is questionable because it degrades the quality of alignment and homology cannot be determined reliably from a lineup.

### 2.3 RAFTS3groups

As an easy way to the user, in order to minimize time and maintaining consistency in data analysis with orthologous proteins, we have as a tool *Rapid Alignment Free Tool for Sequences Similarity Search to Groups* (RAFTS3groups). As the name implies, the tool does not use alignment calculations because it is an alignment free tool Rapid Alignment algorithm-based Free Tool for Sequences Similarity Search (RAFTS3grous) (Vialle, 2013). RAFTS3 (available at https://sourceforge.net/projects/rafts3/) is, succinctly, a tool that looks for similarities between protein sequences and that also uses a filter (*Hash* function) for the selection of candidates based on *k-mers* shared and a measure of comparison using a co-occurrence matrix of amino acid residues (BCOM). RAFTS3groups, was initially developed as an application of the RAFTS3 tool for grouping of homologous sequences and was described earlier in the work of Coimbra (Coimbra, 2015). RAFTS3groups receive as input the file multiFASTA to nucleotides or proteins to be grouped. The script creates a bank in which the query is performed by the RAFTS3 algorithm. Each protein in the input file is evaluated, initially, as a group, and the result of the query to the Bank assesses the similarity between the proteins by the value of self-score. Orthologous groups are formed on the basis of the verification of the number of cases recognised by the minimum similarity assessed (self-score equal to 0.5) (Coimbra, 2015).

The functioning of the algorithm takes place in two main stages: the first is responsible for the bank format and generates *clusters* of proteins using RAFTS3 algorithms. The second stage is the filtering and cluster improvement through analysis of minimum maximum in that group and groups information to calculate k-*means*, are used to better predict the protein *clusters* based on orthology for the implementation of the *DivideCluster* algorithm. We adopted the a threshold 0.5 (or 50%) of self-score (analogous to 50% similarity implementing PAM and BLOSUM matrices - Coimbra, 2015), as a criterion of comparison with UCLUST and CD-HIT. For them it was adopted the similarity threshold of 0.5.

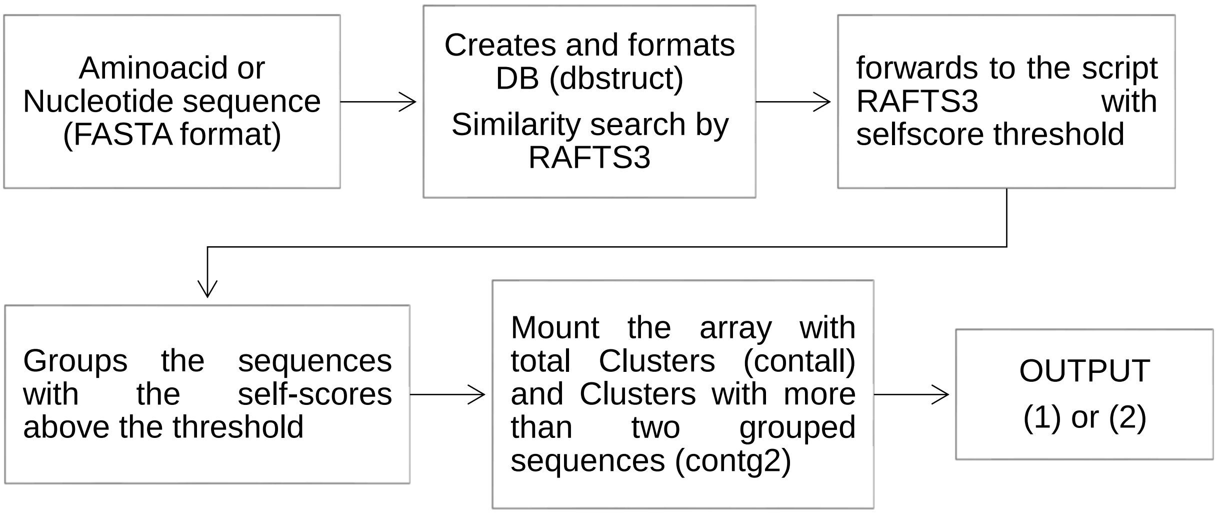
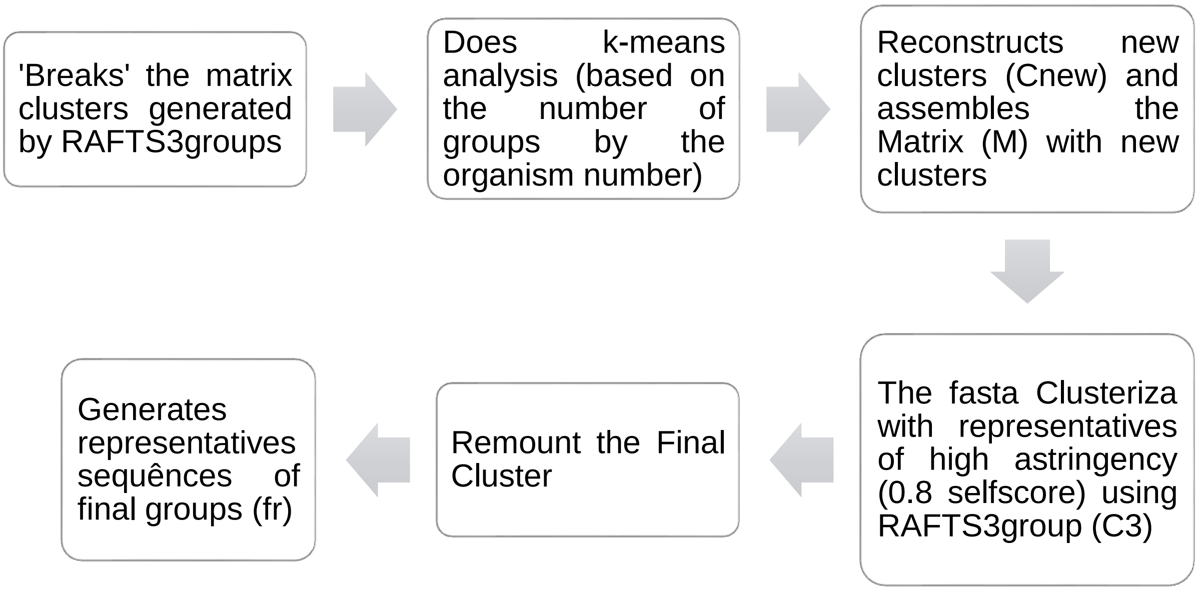

Aminoacid or Nucleotide sequence (FASTA format)
Creates and formats DB (dbstruct)
Similarity search by RAFTS3
forwards to the script RAFTS3 with selfscore threshold
Groups the sequences with the self-scores above the threshold
Mount the array with total Clusters (contall) and Clusters with more than two grouped sequences (contg2)
OUTPUT
(1) or (2)

FIGURE 1 - BASIC WORKFLOW RAFTS3GROUPS ALGORITHM.

Note-If he can have two basic outputs: data generated by the algorithm can be extracted and analyzed (1) or data can receive one more step by the script *DivideCluster* and filtering and follow for homology analysis step (2).



'Breaks' the matrix clusters generated by RAFTS3groups
Does k-means analysis (based on the number of groups by the organism number)
Reconstructs new clusters (Cnew) and assembles the Matrix (M) with new clusters
The fasta Clusteriza with representatives of high astringency (0.8 selfscore) using RAFTS3group (C3)
Remount the Final Cluster
Generates representatives sequências of final groups (fr)

FIGURE 2. BASIC WORKFLOW OF THE STRUCTURE OF THE *DIVIDECLUSTER* ALGORITHM FOR ORTHOLOGS CLUSTERING

The second stage is the filtering and cluster improvement through analysis of minimum maximum in that group and information bodies, k-means, calculation are used for better predict protein *clusters* based on orthology. The first step can easily be crafted without the need of the second step, serving as a tool of clustering, because two different algorithms are used at each step, the first uses the RAFTS3group script and the second uses the *DivideCluster* script. During the second stage, the *clusters* generated by the RAFTS3groups algorithm in the form of array is parsed with groups and information agencies of the initial set of FASTA format, *clusters* are separated and, when number of *clusters* are larger than the number of organisms, a re-clustering is made with the help of calculus k-means clustering. The matrix is rebuilt and a new array is generated with the new *clusters* or reassembled.

## Results and Discussions

## 3.1 Clustering UniProt/Swiss-Prot Bank by RAFTS3groups in Different Self-Scores

One of the ideas and criteria to evaluate the potential of RAFTS3groups was to analyze, in various self-scores, performance profile in relation to time, number of unique *clusters*, representative and total *clusters*. We had adopted as unique or singles *clusters*, those *clusters* with only one sequence by cluster, being itself the representative sequence. Representative *clusters* are groups composed of at least two sequences. And total clusters are made by the representative clusters and unique clusters. We selected seven different self-scores with of threshold 30, 40, 50, 60, 70, 80, 90 against the UniProtKB/Swiss-Prot database. The tool was developed in Matworks environment MATLAB 2012b vers, in OS Biolinux 8 (Ubuntu Linux-based operating system 64-bit 14.04 LTS released and made available in July 2014). The hardware specifications was in a Lenovo Desktop with core i5-650 quad-core 3.20 Giga Hertz (GHz) and 12 Gigabytes (Gb) of RAM. We observed that the clusterization time was directly proportional to the increase in selfscore. Having minimum clustering of about 6.3 hours in 30 selfscore threshold and the time spent on selfscore clustering with 90 total time of 13.6 hours. It was noted also that the average increase of unique *clusters* increased by 0.23 and 0.13 in representative *clusters* and *clusters* of approximately 0.20 on total *clusters*. In relation to the time the tool kept an average of 10.75 hours, as shown in TABLE 1 and FIGURE 3.

TABLE 1. CLUSTERING RESULT BY RAFTS3GROUPS USING VARIOUS SELFSCORE METRICS

| SelfScore | ***Unique*** **Clusters** | ***Representative*** **Clusters** | ***Total*** **Clusters** | ***Analysis Time (in Hours) **** |
|---|---|---|---|---|
| **30** | 55 042 | 34 399 | 89 441 | 6.3 |
| **40** | 64 082 | 39 759 | 103 841 | 9.6 |
| **50** | 77 357 | 46 848 | 124 205 | 10.5 |
| **60** | 97 536 | 55 268 | 152 804 | 11.1 |

| | | | | |
|---|---|---|---|---|
| **70** | 126 840 | 63 551 | 190 391 | 11.5 |
| **80** | 168 754 | 69610 | 238 364 | 12.7 |
| **90** | 231 573 | 70 589 | 302 162 | 13.6 |

Number of *clusters* obtained in different self-livescore by RAFTS3groups algorithm in MATLAB Matworks v2012b in BioLinux operating system vers 8 OS.

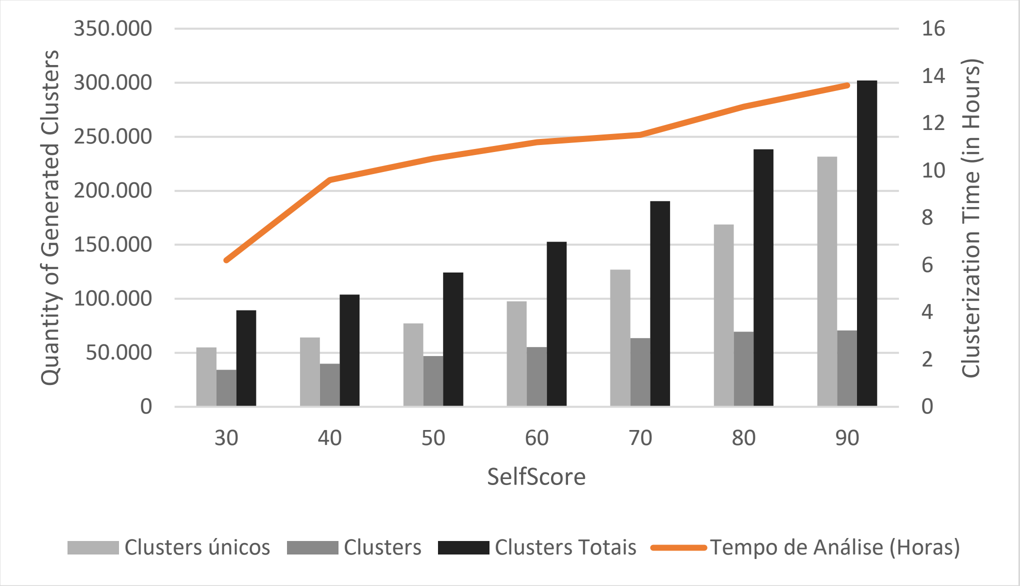


FIGURE 3. Unique, Representative and Total *Clusters* Generated By the RAFTS3groups tool In Different Self-Scores.  Relationship between time in hours of analysis (the vertical line on the right), number of *Clusters* generated (vertical line) in 7 different self-scores (horizontal line).

* The value includes, besides the clustering algorithm, the formatting of the binary matrix by formatdb script.

Although simple, data generated with preliminary analyses with the use of RAFTS3groups may appear on clustering tool efficiency, because, for example, the cluster segmentation as the rise of self-score proceeds, since, if the threshold is increased, less expected the number of sequences that are close and categorized within a same cluster.

### *3.2 Comparison between diferent clustering tools*

We choose to test and evaluate some parameters such as the speed, the amount of *clusters* and data generated by the outputs of the CD-HIT algorithms (version 4.6.1), RAFTS3group and UCLUST (version 8.1.1861) in order to validate the *clusters* generated by our proposal for clustering tool. For such a task, all software were shot in Biolinux 8 operating system (based on Ubuntu Linux 14.04 LTS 64-bit and released in july 2014). The ability of hardware was in a Lenovo Desktop with core i5-650 3.20 GHz quad-core, 12 GB of RAM.

As a criterion of comparison we used the sequences of biological database UniProtKB/Swiss-Prot collected in February 2016 containing 550,299 sequences of proteins analyzed and manualy cured. The Data Bank was chosen because the volume of information is relatively low and satisfactory for the analysis between the tools, because, if we compare the TrEMBL, the amount of information it would be much higher (more than 64,000,000 sequences deposited until June 2016), and it would take months to accomplish all the results. For all the tools we try for 50 percent identity threshold (or 0.5) and the generated *clusters* were accounted.

We noted that it was the tool that UCLUST more clustering data generated with 126,567, followed by RAFTS3groups with CD-HIT with 119,563 and 124,205 Total *clusters*. However, if in the case of generated *clusters* with at least two sequences, CD-HIT was more sensitive to 79,380 *clusters*, followed by UCLUST with 56,737 and RAFTS3groups with 46,838 *Clusters*. About sequences that have not found satisfactory Hits to generate a new cluster, the RAFTS3group was the most generated data with unique *Clusters* followed by 77,357 UCLUST with 70,102 and CD-HIT with 40,183 *Clusters*. For the time analysis, under the same conditions of Hardware, the CD-HIT algoritm was that it took longer, more than 41 hours, longer than about 5 times and 4 times of that the UCLUST and RAFTS3grous, respectively.

TABLE 2**.** CLUSTERING RESULTS USING CD-HIT, UCLUST AND RAFTS3GROUPS

| ***Metodology*** | ***Run time*** | ***Representative*** | ***Unique*** | ***Total*** | ***Generated*** |
|---|---|---|---|---|---|
| | ***(Hours)*** | **Clusters** | **Clusters** | **Clusters** | ***Data (Kb)*** |
| CD-HIT | 41.2 | 79 380 | 40 183 | 119 563 | 24 111 |

| | | | | | |
|---|---|---|---|---|---|
| UCLUST | 8.0 | 56 737 | 70 102 | 126 567 | 43 172 |
| RAFTS3group | 10.5* | 46 848 | 77 357 | 124 205 | 11 259** |

Results obtained with the Tools CD-HIT, RAFTS3groups and UCLUST using UniProtKB Bank with Swiss-prot protein sequences 550,299 (February 2016) using 50% similarity or 50% self-score.

* The value includes, besides the clustering algorithm, the formatting of the binary array at fomatdb

** The total data generated by the tool is based on your array generated on MATLAB.

We also made analysis of sequence per *cluster* and representative *clusters*. For this we have selected and collected to analysis, randomly of the outputs, sequences within the corresponding *clusters* observed in RAFTS3groups, CD-HIT and UCLUST. For this we have selected at random (see supplementary material) 50 representative *clusters* of RAFTS3groups and filter in the *clusters* generated by CD-HIT and UCLUST, through by the identification mumber (GI) of each string. The study revealed that, RAFTS3groups, was the tool that, compared to CD-HIT and UCLUST, obtained greater overall number of clustered proteins, but it is linear coefficient, approached from the other two tools (0.81 of CD-HIT and 0.97 of UCLUST) as shown in the graph in Figure 4.

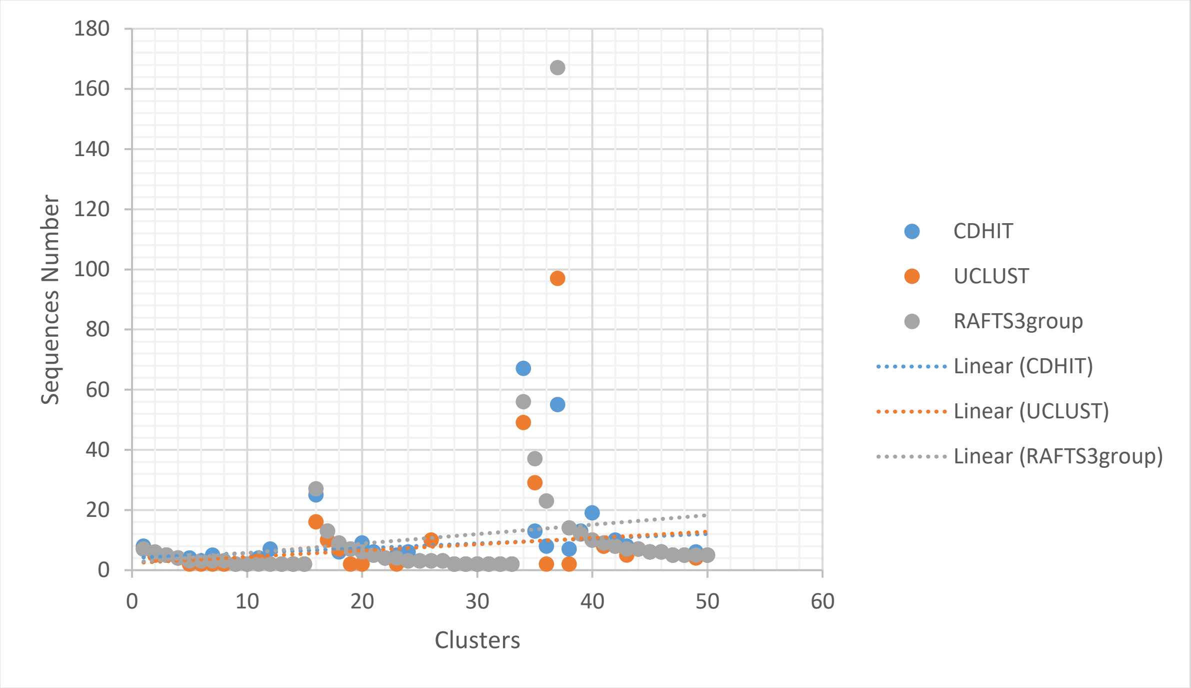


FIGURE 4. NUMBER OF SEQUENCES PER CLUSTER COMPARISON BETWEEN THE CD-HIT, UCLUST AND RAFTS3GROUPS

Result for 50 representative *clusters* analysis for CD-HIT, RAFTS3groups and UCLUST obtained with 50% threshold.This result does not configure that RAFTS3groups is better than or more sensitive, but seem to infer that RAFTS3groups efficiency is comparable to the Tools CD-HIT and UCLUST.

### *3.3 Analysis and comparision with BLAST methodology in obtaining orthologous groups in Herbaspirillum spp.*

One of the strategies used, already mentioned earlier in this work for the analysis of homology is the BLAST ' ‘all-against-all’ ' in set of genes or proteins present in the *Herbaspirillum* spp. assessed genomes. The technique consists in confronting these sets to evaluate how many and which proteins are shared between organisms. This analysis can be displayed in form of matrix of homology, called BLAST matrix. From the BLAST ‘all-against-all’ one can also check the genes common to all genomes (genome core), the same way that the entire genic/proteomic set analyzed (pangenome) (BINNEWIES et al., 2006; LUKJANCENKO et al., 2012).

We have Collected 9 complete genomes of *Herbaspirillum* spp. from National Center for Biotechnology Information (NCBI) Bank that include: *Herbaspirillum* frisingense GSF30, *Herbaspirillum* hiltneri N3, *Herbaspirillum* rubrisubalbicans M1, *Herbaspirillum* seropedicae SmR1, *Herbaspirillum* seropedicae strain Z67, *Herbaspirillum* sp. CF444, *Herbaspirillum* sp. GW103 , *Herbaspirillum* massiliense JC206, *Herbaspirillum* sp. YR522 in FASTA text format, until the date of 5/20/2016.

TABLE 3. GENOME REFERENCES TO ORTHOLOGY STUDIES OBTAINED FROM NCBI DEPOSITED UNTIL 05/20/2016.

| Organism | Version | Accession Number |
|---|---|---|
| Herbaspirillum frisingense GSF30 | AEEC02000093.1 GI:481866950 | AEEC02000093 |
| Herbaspirillum hiltneri N3 | CP011409.1 GI:917675518 | CP011409 |
| Herbaspirillum rubrisubalbicans M1 | CP013737.1 GI:971149481 | CP013737 |
| Herbaspirillum seropedicae SmR1 | CP002039.1 GI:300072131 | CP002039 |
| Herbaspirillum seropedicae strain Z67 | CP011930.1 GI:852454696 | CP011930 |
| Herbaspirillum sp. CF444 | AKJW01000001.1 GI:398104681 | AKJW01000001 |
| Herbaspirillum sp. GW103 | AJVC01000001.1 GI:386435888 | |
| Herbaspirillum massiliense JC206 | NZ_HE978634.1 GI:484029580 | NZ_HE978634 |
| Herbaspirillum sp. YR522 | AKJA01000001.1 GI:398224158 | AKJA01000001 |

We extracted a total of 41,963 protein sequences (4,663 average by proteins genome). The results obtained with the sequences using BLAST strategy ' ‘all-against-all’ ' and their results (described and made available also on the work of Cardoso, 2015 - not published) were confronted with the results obtained by the RAFTS3groups tool, employing orthology clustering algorithm *DivideCluster*. The results were filtered and are described immediately below in TABLE 4.

TABLE 4. UNIQUE AND CORE SEQUENCES TO *HERBASPIRILLUM* SPP. USING BLAST ALGORITHM ' ALL-AGAINST-ALL ' AND RAFTS3GROUPS

| | **BLAST** | **BLAST** | **RAFTS3groups** | **RAFTS3groups** | **Total** |
|---|---|---|---|---|---|
| | **Unique** | **Core** | **Unique** | **Core Sequences** | **proteins** |
| **Organism** | **Sequences** | **Sequences** | **Sequences** | | |

| | | | | | |
|---|---|---|---|---|---|
| ***Herbaspirillum frisingense* GSF30** | 456 | 4415 | 517 | 4354 | 4871 |
| ***Herbaspirillum hiltneri* N3** | 402 | 3925 | 536 | 3791 | 4327 |
| ***Herbaspirillum rubrisubalbicans* M1** | 727 | 3965 | 515 | 4177 | 4692 |
| ***Herbaspirillum seropedicae* SmR1** | 203 | 4534 | 242 | 4495 | 4737 |
| ***Herbaspirillum seropedicae* strain Z67** | 102 | 4563 | 83 | 4582 | 4665 |
| ***Herbaspirillum* sp. CF444** | 530 | 4444 | 985 | 3989 | 4974 |
| ***Herbaspirillum* sp. GW103** | 430 | 4225 | 541 | 4114 | 4655 |
| ***Herbaspirillum* massiliense JC206** | 1738 | 2692 | 2048 | 2382 | 4430 |
| ***Herbaspirillum* sp. YR522** | 810 | 3802 | 1087 | 3525 | 4612 |
| TOTAL | **5398** | **36565** | **6554** | **35409** | **41963** |

Compared with the corresponding *clusters* obtained a level of 96%. Of the analyzed sequences 41963, the BLAST returned about 87.14% of cured proteins as sequences contained in all organisms studied and only 12.86% presented as unique sequences of each species. The RAFTS3groups algorithm came up much of these values. About of 84.40% of sequences, were core sequences between organisms studied. With only 15.6% of proteins unique to one or the other species. We also obtained a degree of correlation of 96% to 95% and core sequences for unique sequences, that be showned in FIGURE 5.

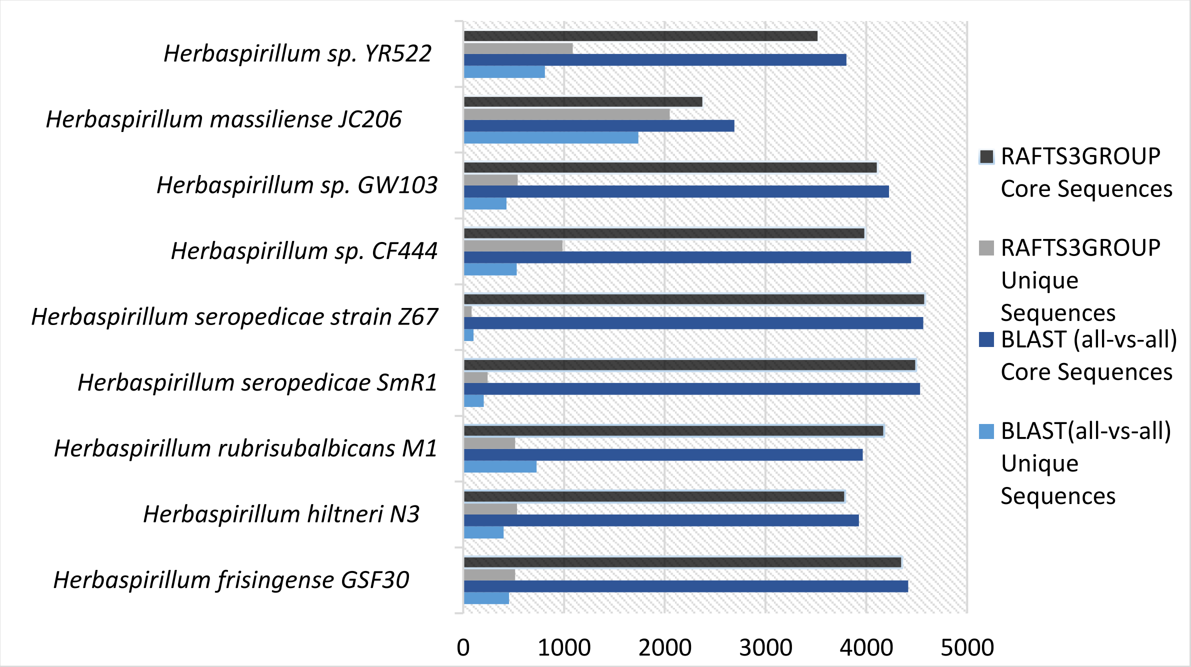


FIGURE 5. COMPARATIVE ANALYSIS BETWEEN COUNTERPARTS IN BLAST 'ALL-AGAINST-ALL' AND RAFTS3GROUPS

The chart presents results obtained in clustering of orthologs using the methodology BLAST all-against-all and RAFTS3groups. Was analyzed the amount of core and unique sequences sequences of both methodologies, being observed a strong correlation (96%) between the data volume.

## 4 Conclusion

The goal of this work was to bring a new fast and efficient tool for the analysis of *clusters* involving large volume of data and also introducing a new clustering approach orthologs clustering using the K-means algorithm.

We used the methodology CD-HIT because it is a tool that aims to work with large amounts of biological data and UCLUST for being a tool of clustering in the same category as RAFTS3groups. With the obtained results we can infer that our technique is as efficient as the two and their agility comparable to UCLUST.

In homology analysis, the methodology BLAST everyone against everyone is unanimous in orthology analysis between sequences, but it is a technique already described that requires a large computational demand that hinders the fluidity of several studies. As we develop the RAFTS3groups algorithm, a fast algorithm and that maintains consistency in the groups generated, because it equates the methodology of BLAST ‘all-against-all’.

We hope to contribute positively to new studies involving homology between sequences, bringing a new alternative to clustering and consolidation of orthologs groups, which is very important to comparative genomics and bioinformatics studies.

## 5 Acknowledgements

CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and Fundação Araucária supported this work.

## 6 Conflicts of interest

No potential conflict of interest relevant to this article was reported

# 7 References

ALTHENHOFF, A. M.; & DESSIMOZ, C. (2009). Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods. *PLoS Computational Biology*, *5*(1), e1000262.

APWEILER, R., BAIROCH, A., WU, C. H., BARKER, W. C., BOECKMANN, B., FERRO, S., … Yeh, L.-S. L. (2004). UniProt: the Universal Protein knowledgebase. *Nucleic Acids Research*, *32*(Database issue), D115–9.

BINNEWIES, T. T.; MOTRO, Y.; HALLIN, P. F.; et al. (2006). Ten years of bacterial genome sequencing: comparative-genomics-based discoveries. Functional & Integrative Genomics, v. 6, n. 3, p. 165–185.

BITARD-FEILDEL, T.; KEMENA, C.; GREENWOOD, J. M.; & BORNBERG-BAUER, E. (2015). Domain similarity based orthology detection. *BMC Bioinformatics*, *16*(1), 154.

CARDOSO, R. L. A. (2015). Análise genômica comparativa de bactérias do gênero *Herbaspirillum*.

CHEN, F., MACKEY, A. J., VERMUNT, J. K., & ROOS, D. S. (2007). Assessing Performance of Orthology Detection Strategies Applied to Eukaryotic Genomes. *PLoS ONE*, *2*(4), e383.

CHEN, T., WU, T. H., NG, W. V, & LIN, W. (2010). DODO: an efficient orthologous genes assignment tool based on domain architectures. Domain based ortholog detection. *BMC Bioinformatics*, *11 Suppl 7*(Suppl 7), S6. http://doi.org/10.1186/1471-2105-11-S7-S6

COIMBRA, N. A. R. Metodologia Computacional para o Estudo de Genes com vizinhança conectada: Análise do cluster nif
genômica. Dissertação (Mestrado em Bioinformática) – Setor de Educação Profissional e Técnológica, Universidade Federal do Paraná, Curitiba, 2015.

CURTIS, D. S., PHILLIPS, A. R., CALLISTER, S. J., CONLAN, S., & MCCUE, L. A. (2013). SPOCS: software for predicting and visualizing orthology/paralogy relationships among genomes. *Bioinformatics*, *29*(20), 2641–2642.

DALQUEN, D. A., ALTENHOFF, A. M., GONNET, G. H. and DESSIMOZ, C. (2013) The impact of gene duplication, insertion, deletion, lateral gene transfer and sequencing error on orthology inference: a simulation study. PloS One, 8, e56925.

EDGAR, R. C. (2010) Search and clustering orders of magnitude faster than BLAST, *Bioinformatics* 26(19), 2460-2461.

EMMS, D. M., & KELLY, S. (2015). OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biology*, *16*(1), 157.

FITCH W. M. (2000). Homology a personal view on some of the problems. Trends Genet 16: 227–31

FREEMAN, S. & HERRON, J., (2009) C. *Análise Evolutiva* - 4ª Edição: Porto Alegre: ArtMed Editora.

HUANG, Y., NIU, B., GAO, Y., FU, L., & Ii, W. (2010). CD-HIT Suite: A web server for clustering and comparing biological sequences. *Bioinformatics*, *26*(5), 680–682.

JENSEN R. A. (2001). Orthologs and paralogs - we need to get it right. *Genome Biology*. 2(8): interactions1002.1-interactions1002.3.

KIM, K., KIM, W., & KIM, S. (2011). ReMark: An automatic program for clustering orthologs flexibly combining a Recursive and a Markov clustering algorithms. *Bioinformatics*, *27*(12), 1731–1733.

KOONIN E.V. (2005). "Orthologs, paralogs, and evolutionary genomics". *Annual Review of Genetics* 39: 309–38.

KRISTENSEN, D. M., WOLF, Y. I., MUSHEGIAN, A. R., & KOONIN, E. V. (2011). Computational methods for Gene Orthology inference. *Briefings in Bioinformatics*, *12*(5), 379–391. http://doi.org/10.1093/bib/bbr030

KUZNIAR, A., VAN HAM, R. C. H. J., PONGOR, S., & LEUNISSEN, J. A. M. (2008). The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics*, *24*(11), 539–551.

LECHNER, M.; FINDEIB, S., STEINER, L., MARZ, M., STADLER, P. F., & PROHASKA, S. J. (2011). Proteinortho: Detection of (Co-)orthologs in large-scale analysis. *BMC Bioinformatics*, *12*(1), 124.

LI, W., & Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, *22*(13), 1658–1659.

LINARD, B. (2011). OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, *12*(11), 1471.

LUKJANCENKO, O.; USSERY, D. W.; WASSENAAR, T. M. (2012). Comparative Genomics of *Bifidobacterium***,** *Lactobacillus* and Related Probiotic Genera. Microbial Ecology, v. 63, n. 3, p. 651–673.

MAFRA, T.; Bioinformaticando – Praticando um pouco de Bioinformática. Disponível em: http://bioinformaticando.blogspot.com.br/2012/04/orthomcl-busca-por-genes-ortologos.html?view=sidebar, 2012 Access in: 05/03/2016.

COIMBRA, N. A. R. (2015). Metodologia Computacional para o Estudo de Genes com vizinhança conectada: Análise do cluster nif
genômica. Dissertação (Mestrado em Bioinformática) – Setor de Educação Profissional e Técnológica, Universidade Federal do Paraná, Curitiba.

PROSDOCIMI F. (2007). Curso Online de Introdução à Bioinformática, available in: http://www2.bioqmed.ufrj.br/prosdocimi/FProsdocimi07_ CursoBioinfo.pdf.

VIALLE, A. R. (2013). SILA – Ferramenta de alto desempenho para anotação automática genômica. Dissertação (Mestrado em Bioinformática) – Setor de Educação Profissional e Técnológica, Universidade Federal do Paraná, Curitiba.

WAGNER, I.; VOLKMER, M.; SHARAN, M.; VILLAVECES, J. M.; OSWALD, F.; SURENDRANATH, V.; & HABERMANN, B. H. (2014). MorFeus: a web-based program to detect remotely conserved orthologs using symmetrical best hits and orthology network scoring. *BMC Bioinformatics*, *15*(1), 263.

WANG, Y.; COLEMAN-DERR, D.; CHEN, G.; & GU, Y. Q. (2015). OrthoVenn: a web server for genome wide comparison and annotation of orthologous *clusters* across multiple species. *Nucleic Acids Research*, *43*(W1), W78–W84.

ZAHA, A.; FERREIRA, H. B.; PASSAGLIA, L. M. P. (Org.). Biologia molecular básica. 5. ed. Porto Alegre: Artmed, 2014.  P 416.

# 4 RESULTADOS

## 4.1 FERRAMENTAS DE ORTÓLOGOS

### 4.1.1 A PERSISTÊNCIA DO ALGORITMO BLAST

Após analisarmos todas as 14 ferramentas de ortólogos, observamos alguns aspectos importantes e alguns problemas que ainda persistem na criação de grupos. Apesar de termos várias ferramentas para criação ou análise de ortologia disponíveis, cada uma atende a um problema específico dentro do contexto de ortologia. Poucas são as ferramentas que conseguem atingir um amplo espectro de organismos ou de sequências, por exemplo. Uma problemática que também persiste e que limita a solução de alguns problemas na criação de grupos de ortólogos é a utilização do algoritmo BLAST em alguma estapa do processo. Poucas ferramentas trazem algoritmos inovadores, ou alternativas ao BLAST, sendo que um dos maiores gargalos da bioinformática a ser enfrentado, é a demanda de gasto computacional e de tempo necessário para a clusterização por parte dos algoritmos. Pressupõe-se que, a persistência em se utilizar o BLAST, é pelo fato de ser padrão ouro em análises de clusterização de ortólogos e de ainda não se ter uma nova abordagem que seja tão eficiente quanto o padrão ouro e que o substitua.

### 4.1.2 FERRAMENTAS DE ORTÓLOGOS: AVANÇOS E DESAFIOS

Analisando o perfil de cada ferramenta ou software durante nosso *review,* notamos que, de maneira geral, as ferramentas possuem algumas necessidades para execução localmente em máquinas, como dependência de várias bibliotecas para a execução dos algoritmos, a usabilidade ainda é um pouco complicada, por exemplo na execução da ferramenta Ortholog-Finder, que é necessário a instalação de outros pacotes como BLAST+, OrthoMCL, BioPERL. Em contraste, para superarmos essa dificuldade, temos a ferramenta ProteinOrtho, que apesar da necessidade de algumas dependências computacionais, compromete em minimizar gastos com a memória RAM em uma máquina, reduzindo gasto com supercomputadores, utilizando uma linha de comando mais intuitiva. Outra ferramenta que se destaca por sua inovação dentre as demais, é a ferramenta OrthoVenn, que explora a visualização em forma de diagramas e isso pode facilitar o entendimento de resultados dos grupos, e não simplesmente restrito a um *output* em arquivo texto, por exemplo. Também é importante destacarmos que, quanto mais automático o processo da criação de grupos e de análise, melhor é para o pesquisador e, por isso, as ferramentas SPOCS e ReMark para nós, se destacaram nessa classe de ferramentas. SPOCS é uma ferramenta interessante pois possue tanto interface web quanto a análise pode ser feita localmente, o que não se encontra muito nas demais ferramentas, porém é necessário cadastro no site do seu desenvolvedor para que se receba por e-mail o resultado de alguma análise.

## 4.2 ANÁLISE DE CLUSTERIZAÇÃO DE RAFTS3GROUPS E SEU POTENCIAL PARA ANÁLISES DE ORTOLOGIA

### 4.2.1 CLUSTERIZAÇÃO DO BANCO UNIPROT/SWISS-PROT POR RAFTS3GROUPS EM DIFERENTES *SELFSCORES*

Observou-se que o tempo foi diretamente proporcional ao aumento de *self-score*. Tendo tempo mínimo de clusterização de cerca de 6,3 horas em *selfscore* de limiar 30 e o tempo maior gasto na clusterização de *selfscore* 90 com tempo total de 13,6 horas. Notou-se também que o aumento médio de *clusters* únicos aumentou em 0,23 *clusters* representativos 0,13 e *clusters* totais cerca de 0,20. Em relação ao tempo a ferramenta manteve um tempo médio de 10,75 horas, como ilustrados na FIGURA 7.

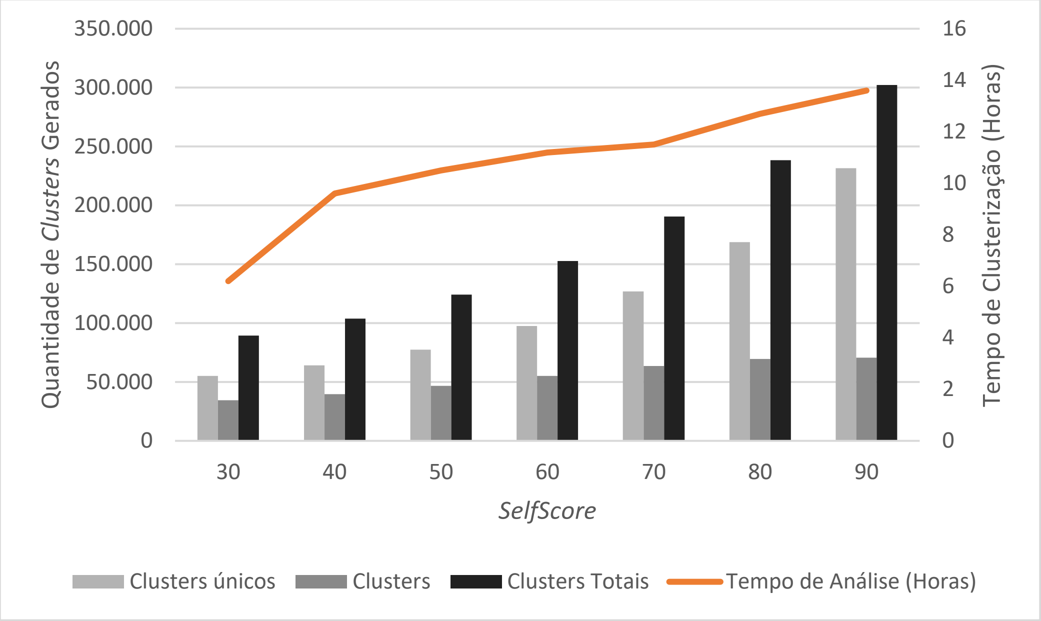


FIGURA 7. *CLUSTERS* ÚNICOS, REPRESENTATIVOS E TOTAIS GERADOS PELA FERRAMENTA RAFTS3GROUP EM DIFERENTES *SELFSCORES*
Dados do relacionamento entre tempo em Horas de análise (linha vertical a direita), número de *Clusters* gerados (linha vertical a esquerda) em 7 *self-scores* diferentes (linha horizontal).
* O valor inclui, além da clusterização do algoritmo, a formatação da matriz binária pelo fomatdb

Apesar de simples, dados gerados com análises preliminares com o uso do RAFTS3groups podem figurar a eficiência da ferramenta na clusterização, pois, por exemplo, a segmentação de *clusters* conforme o aumento de *self-score* procede, tendo em vista que, se o limiar é aumentado, menor se espera o número de sequências que sejam próximas e categorizado dentro de um mesmo *cluster*.

## 4.2.2 CLUSTERIZAÇÃO E ANÁLISE DE RESULTADOS OBTIDOS POR CD-HIT, UCLUST E RAFTS3GROUPS

Optamos por testar e avaliar alguns parâmetros como a velocidade, a quantidade de *clusters* e de dados gerados pelos *outputs* dos algoritmos do CD-HIT (na versão 4.6), do UCLUST (versão 8.1.1861) e do RAFTS3group, afim de validar os *clusters* gerados pela nossa proposta de ferramenta de clusterização. Para tal tarefa, todos os softwares foram também rodados em sistema operacional Biolinux 8 com capacidade de hardware de um Desktop Lenovo core i5-650 de 4 núcleos com *overclock* de 3,20 GHz e 12Gb de memória RAM.

Como critério de comparação utilizamos as sequências do Banco de dados biológico UniProtKB Swiss-Prot, coletado em fevereiro de 2016 contendo 550.299 sequências de proteínas analisadas e curadas manualmente. O banco foi escolhido pelo fato do volume de informações ser relativamente baixo e satisfatório para a análise entre as ferramentas, pois, se compararmos ao próprio TrEMBL, o volume de informações seria muito superior (possui mais de 64 milhões de sequências depositadas até junho de 2016), e seriam necessários meses para realizar todos os resultados. Para todas as ferramentas optamos pelo limiar de identidade de 50% (ou 0.5) e os *clusters* gerados foram contabilizados.

Observou-se que UCLUST foi a ferramenta que mais gerou dados de clusterização com 126.567, seguido por RAFTS3groups com 124.205 e CD-HIT com 119.563 *clusters* totais. Porém, se tratando de *clusters* gerados com pelo menos duas sequências, CD-HIT foi mais sensível com 79.380 *clusters*, seguido por UCLUST com 56.737 e RAFTS3groups com 46.838 *Clusters*. EM relação a sequências que não encontraram Hits satisfatórios para gerar um novo cluster, o RAFTS3group foi o que mais gerou dados com 77.357 *Clusters* únicos seguido pelo UCLUST com 70.102 e CD-HIT com 40.183. Partindo para a análise de tempo, as três ferramentas nas mesmas condições de Hardware, a ferramenta CD-HIT foi a que levou mais tempo, mais de 41 horas, tempo superior cerca de 5 vezes e 4 vezes que as ferramentas UCLUST e RAFTS3grous, respectivamente.

TABELA 1 – RESULTADO COMPARATIVO ENTRE AS FERRAMENTAS CD-HIT RAFTS3GROUPS E UCLUST

| Metodologia | Tempo de execução (Horas) | *Clusters* | *Clusters* Unicos | *Clusters* Totais | Dados Gerados (Kb) |
|---|---|---|---|---|---|
| CD-HIT | 41,2 | 79.380 | 40.183 | 119.563 | 24.111 |
| UCLUST | 8,0 | 56.737 | 70.102 | 126.567 | 43.172 |
| RAFTS3group | 10,5* | 46.848 | 77.357 | 124.205 | 11.259** |

Utilizado o banco UniProtKB Swiss-prot com 550.299 sequências de proteínas (fevereiro de 2016) utilizando 50% similaridade ou 50% de *self-score*.

*  O valor inclui, além da clusterização do algoritmo, a formatação da matriz binária pelo fomatdb
**O total de dados gerados pela ferramenta é baseado em sua matriz gerada via MATLAB.

Realizamos também uma análise de números de sequências dentro de *clusters* correspondentes. Para isso selecionamos, aleatoriamente, 50 *clusters* representativos de RAFTS3groups e, selecionando outros 50 *clusters* de CD-HIT e UCLUST excluindo-se os *clusters* únicos. A seleção, baseou-se em elencar uma sequência representativa de RAFTS3groups e filtrar nos *clusters* gerados por CD-HIT e UCLUST, através do número de identificação contido (GI) contido na *Header* de cada sequência. Os resultados obtidos podem ser verificados na TABELA 4 (material suplementar).

O estudo revelou que, RAFTS3groups, foi a ferramenta que, comparativamente ao CD-HIT e UCLUST, obteve maior número geral de proteínas clusterizadas, porém seu coeficiente linear, aproximou-se das outras duas ferramentas (0,81 de CD-HIT e 0,97 de UCLUST) conforme demonstrado no gráfico da FIGURA 8.

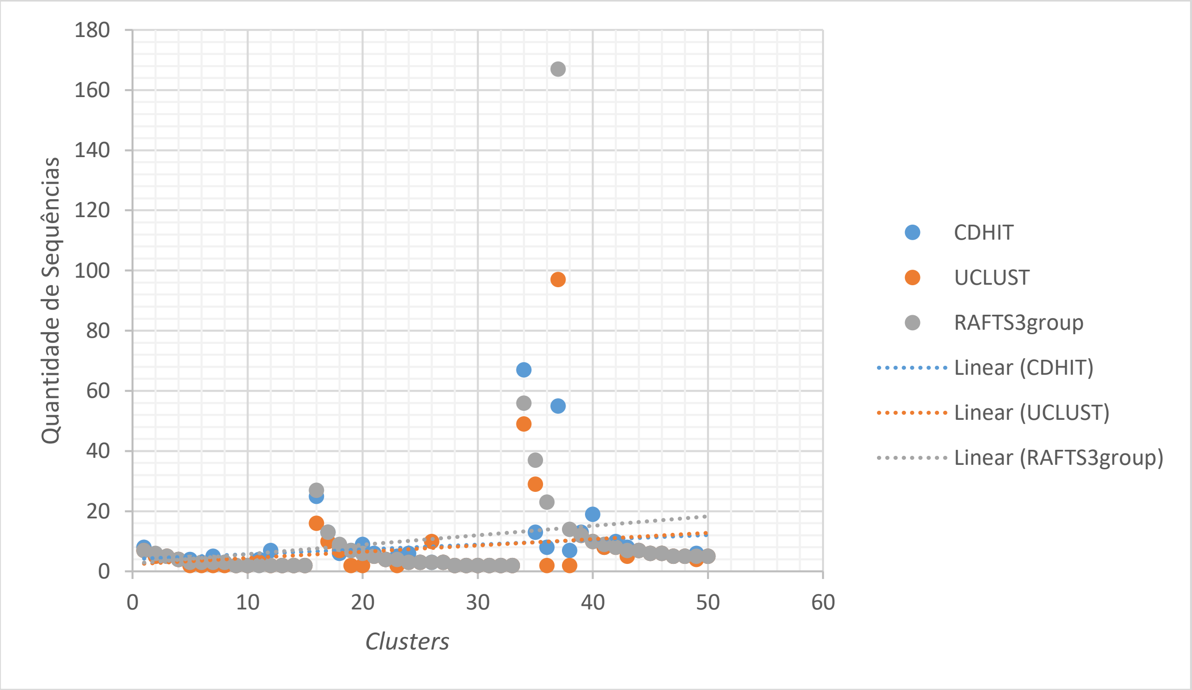


FIGURA 8. COMPARATIVO ENTRE AS FERRAMENTAS CD-HIT, UCLUST e RAFTS3GROUPS

Resultado para análise de 50 *clusters* representativos para CD-HIT, RAFTS3groups e UCLUST obtidos com limiar de 50% de identidade. Esse resultado não figura que RAFTS3groups seja superior ou mais sensível, porém busca inferir que a eficiência de RAFTS3groups é equiparável as ferramentas CD-HIT e UCLUST.

### 4.2.3 COMPARAÇÃO DA METODOLOGIA BLAST E RAFTS3GROUPS NA OBTENÇÃO DE ORTÓLOGOS EM *HERBASPIRILLUM* SPP.

Uma das estratégias utilizadas, já referida anteriormente no trabalho, para análise de homologia é o BLAST 'todos contra todos' (All-against-all BLAST, do inglês) em conjunto de genes ou proteínas presentes nos genomas avaliados de *Herbaspirillum* spp.. A técnica consiste em confrontar esses conjuntos para avaliar quantas e quais proteínas são compartilhadas entre os organismos. Essa análise pode ser visualizada em forma de matriz de homologia, chamada de matriz BLAST. A partir do BLAST todos contra todos também é possível verificar os genes comuns a todos os genomas (core genoma), bem como todo o conjunto gênico/proteômico analisado (pangenoma) (BINNEWIES et al., 2006; LUKJANCENKO et al., 2012).

Para a análise, coletamos 9 genomas completos de *Herbaspirillum* spp. do Banco NCBI (*Herbaspirillum* frisingense GSF30 (version AEEC02000093.1 GI:481866950), *Herbaspirillum* hiltneri N3 (version CP011409.1 GI:917675518), *Herbaspirillum* rubrisubalbicans M1 (version CP013737.1 GI:971149481), *Herbaspirillum* seropedicae SmR1 (version CP002039.1 GI:300072131), *Herbaspirillum* seropedicae strain Z67 (version CP011930.1 GI:852454696), *Herbaspirillum* sp. CF444 (version AKJW01000001.1 GI:398104681), *Herbaspirillum* sp. GW103 (version AJVC01000001.1 GI:386435888), *Herbaspirillum* massiliense JC206 (version NZ_HE978634.1 GI:484029580), *Herbaspirillum* sp. YR522 (version AKJA01000001.1 GI:398224158)) em formato texto FASTA, até a data de 20/05/2016. Extraímos um total de 41.963 sequências proteicas (média de 4.663 proteínas por genoma). Os resultados obtidos com as sequências utilizando a estratégia de BLAST 'todos contra todos' e seus resultados (descritos e disponibilizados no trabalho de CARDOSO, 2015 - não publicado) foram confrontados com os resultados obtidos pela ferramenta RAFTS3groups, empregando o algoritmo de clusterização para ortologia *DivideCluster*. Os resultados obtidos foram filtrados e estão descritos logo adiante na TABELA 2.

TABELA 2. SEQUÊNCIAS UNICAS E CORE PARA *HERBASPIRILLUM* SPP. UTILIZANDO ALGORITMO BLAST 'TODOS-CONTRA-TODOS' E RAFTS3GROUPS

| Organismo | BLAST *Sequências Únicas* | BLAST *Sequências Core* | RAFTS3group *Sequências Unicas* | RAFTS3group *Sequências Core* | Total Proteínas |
|---|---|---|---|---|---|
| *Herbaspirillum frisingense GSF30* | 456 | 4415 | 517 | 4354 | 4871 |
| *Herbaspirillum hiltneri N3* | 402 | 3925 | 536 | 3791 | 4327 |
| *Herbaspirillum rubrisubalbicans M1* | 727 | 3965 | 515 | 4177 | 4692 |
| *Herbaspirillum seropedicae SmR1* | 203 | 4534 | 242 | 4495 | 4737 |
| *Herbaspirillum seropedicae strain Z67* | 102 | 4563 | 83 | 4582 | 4665 |
| *Herbaspirillum sp. CF444* | 530 | 4444 | 985 | 3989 | 4974 |
| *Herbaspirillum sp. GW103* | 430 | 4225 | 541 | 4114 | 4655 |
| *Herbaspirillum massiliense JC206* | 1738 | 2692 | 2048 | 2382 | 4430 |
| *Herbaspirillum sp. YR522* | 810 | 3802 | 1087 | 3525 | 4612 |
| **TOTAL** | **5398** | **36565** | **6554** | **35409** | **41963** |

Comparativamente os *clusters* de homólogos obtivemos um grau de 96% de correspondência. Das 41963 sequências analisadas, o BLAST retornou cerca de 87,14% das proteínas curadas como sequências contidas em todos os organismos estudados e apenas 12,86% apresentaram-se como sequências exclusivas de cada espécie. O algoritmo RAFTS3groups aproximou-se muito desses valores. Cerca de 84,40% das sequências, eram sequências *core* entre os organismos estudados. Restando apenas 15,6% de proteínas exclusiva de uma ou outra espécie. Também obtivemos um grau de correlação de 96% para sequências *core* e 95% para sequências únicas.

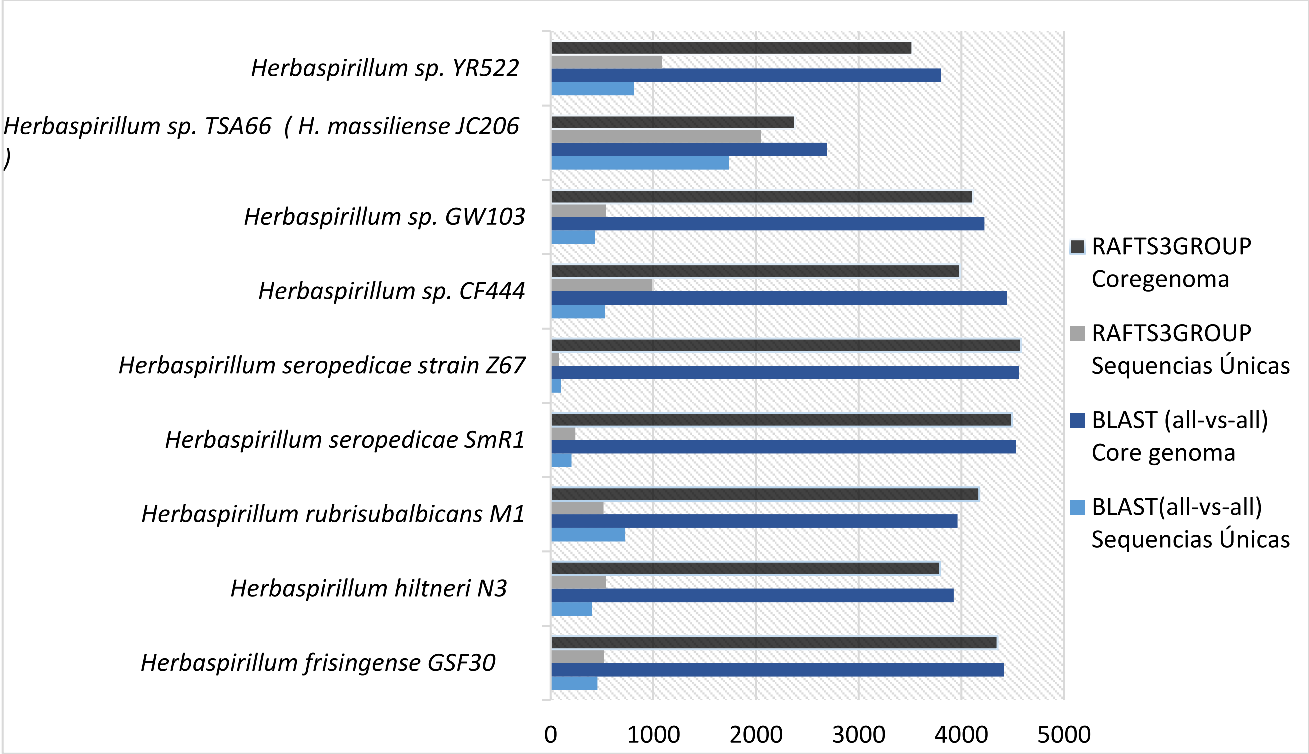


FIGURA 9. ANÁLISE COMPARATIVA ENTRE HOMÓLOGOS DE BLAST ‘TODOS CONTRA TODOS’ E RAFTS3GROUPS

O gráfico apresenta resultados obtidos na clusterização de ortólogos empregando a metodologia BLAST todos-contra-todos e RAFTS3groups. Foi analisado a quantidade de sequências core e sequências únicas de ambas metodologias, sendo observável uma forte correlação (96%) entre o volume de dados.

# 5 CONCLUSÕES

- Mesmo com ferramentas sendo desenvolvidas e aprimoradas, ainda há um forte apelo na obtenção de técnicas ou de metodologias mais rápidas e eficientes na predição de ortólogos, como os descritos no Review. Novas abordagens na criação de grupos, por exemplo, ainda são necessárias em substituição às metodologias que utilizam o BLAST como padrão. Porém, dependendo da necessidade do pesquisador, algumas ferramentas notaram-se promissoras, como PorteinOrtho, que minimiza gasto computacional relacionado ao uso da memória RAM, orthoVenn que traz uma nova abordagem de visualização, utilizando diagrama de Venn para melhor explorar os resultados de grupos ou a minimização da intervenção manual dentro do processo geral de clusterização, tornando-o mais automático, como em SPOCS e ReMark.

- Como uma nova estratégia de clusterização *alignment-free*, a fim de minimizar a grande demanda computacional, lidar com grande volume de dados, otimizando tempo e trazendo uma nova alternativa em substituição ao algoritmo BLAST no campo de clusterização de ortólogos, trouxemos também a validação e a consolidação da ferramenta RAFTS3groups. Tivemos um desempenho muito próximo às ferramentas de clusterização já consolidadas, como o caso do CD-HIT e UCLUST na clusterização de grupos, tendo um desempenho bem superior à técnica empregada na geração do grande banco de proteínas Uniref (cerca de 4 vezes mais rápido), mesmo lidando com grande volume de dados.

- A ferramenta RAFTS3groups também se mostrou eficiente na detecção de grupos de ortólogos com o auxílio da aplicação *DivideCluster,* complementar a ferramenta RAFTS3group, gerando grupos tão próximos quanto a metodologias consolidadas. A ferramenta foi comparada aos obtidos com a metodologia BLAST ‘todos contra todos’ em 9 genomas de *Herbaspirillum* spp. descritos por CARDOSO, 2015, obtendo uma alta correlação entre os dados (96% dos dados grupos de core e pan genomas entre as espécies analisadas).

# 6 MATERIAL SUPLEMENTAR

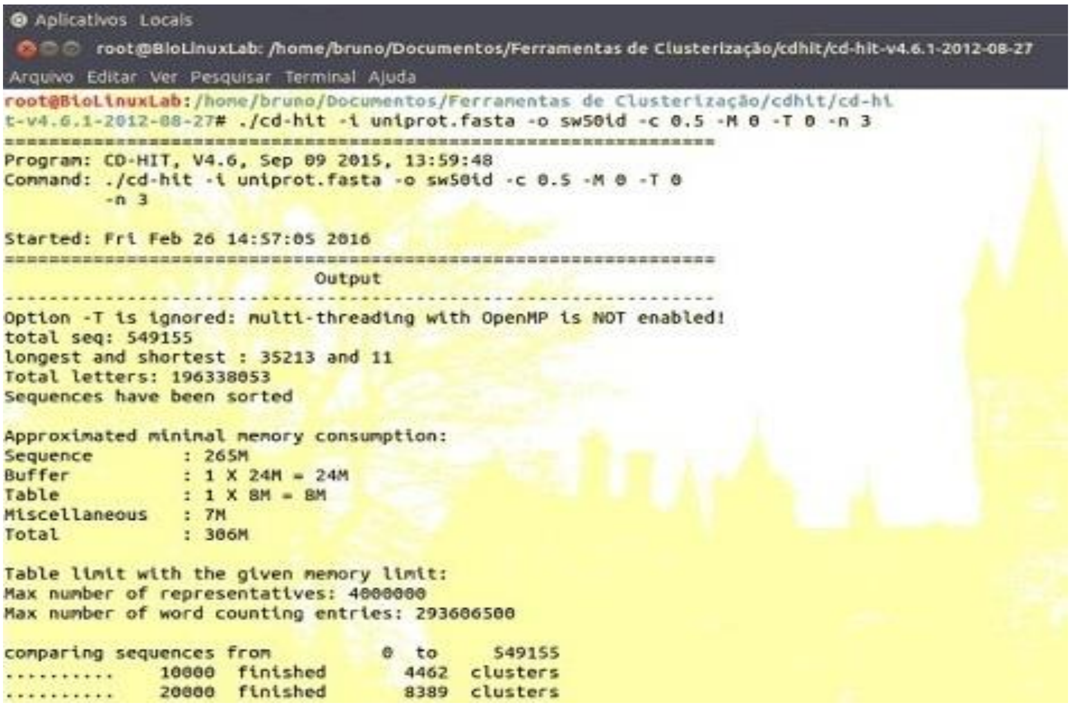
A idéia dessa seção é a de tirar dúvidas referentes aos resultados ou metodologias ou de ferramentas empregadas e descritas durante este trabalho. Arquivos de saídas, tabelas complementares ou dados externos estão contidos nessa seção para melhor facilitar o entendimento do trabalho geral.

## 6.1 RESULTADOS DAS FERRAMENTAS E ARQUIVO DE SAÍDA

### 6.1.1 Resultados do algoritmo CD-HIT

Cada ferramenta tem um formato tabular padrão de porém contendo informações de clusterização diferentes. O *output* é importante para a análise de *clusters*, pois nele estão contidas informações muito importantes ao usuário. Um output pode ter seu conteúdo muito detalhado, porém pobre em informações ou muito sucinto, porém com todas informações básicas ao pesquisador e imprescindíveis para posteriores análises.

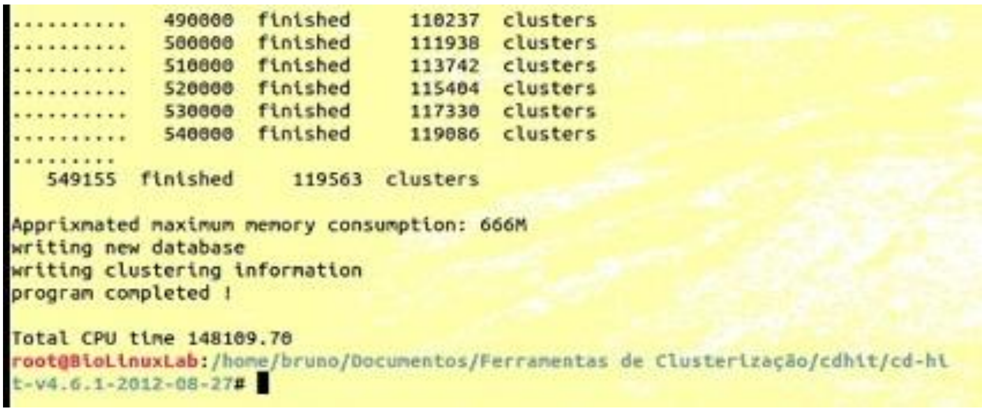
O algoritmo CD-HIT usado localmente, possui uma interface fácil (Figuras 10 e 11) e um arquivo de saída formato de texto em extensão. clstr bastante preciso, com informações de *clusters*, quantidade de aminoácidos por sequencia e porcentagem de identidade em relação à sequencia representativa e geradora do cluster,conforme ilustrado na FIGURA 12.


Aplicativos Locais
root@BioLinuxLab: /home/bruno/Documentos/Ferramentas de Clusterização/cdhit/cd-hit-v4.6.1-2012-08-27
Arquivo Editar Ver Pesquisar Terminal Ajuda

```
root@BioLinuxLab:/home/bruno/Documentos/Ferramentas de Clusterização/cdhit/cd-hi
t-v4.6.1-2012-08-27# ./cd-hit -i uniprot.fasta -o sw50id -c 0.5 -M 0 -T 0 -n 3
================================================================
Program: CD-HIT, V4.6, Sep 09 2015, 13:59:48
Command: ./cd-hit -i uniprot.fasta -o sw50id -c 0.5 -M 0 -T 0
         -n 3

Started: Fri Feb 26 14:57:05 2016
================================================================
                            Output
----------------------------------------------------------------
Option -T is ignored: multi-threading with OpenMP is NOT enabled!
total seq: 549155
longest and shortest : 35213 and 11
Total letters: 196338053
Sequences have been sorted

Approximated minimal memory consumption:
Sequence        : 265M
Buffer          : 1 X 24M = 24M
Table           : 1 X 8M = 8M
Miscellaneous   : 7M
Total           : 306M

Table limit with the given memory limit:
Max number of representatives: 4000000
Max number of word counting entries: 293606500

comparing sequences from          0  to     549155
..........     10000  finished       4462  clusters
..........     20000  finished       8389  clusters
```

FIGURA 10. LINHA E FORMATO DE EXECUÇÃO DA FERRAMENTA CD-HIT

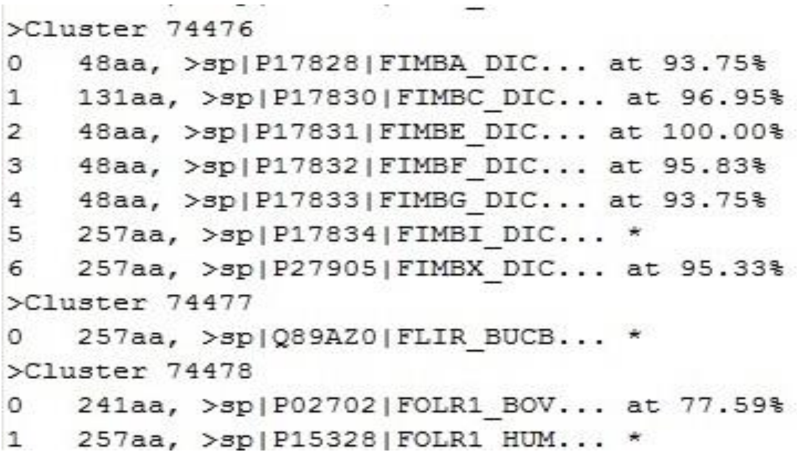
Teste realizado 27/08/2015 exemplificando o comando de execução da ferramenta CD-HIT com limiar de 50% (-id 0.5) aplicado ao Banco de Dados UniProtKB.



```
..........    490000  finished     110237  clusters
..........    500000  finished     111938  clusters
..........    510000  finished     113742  clusters
..........    520000  finished     115404  clusters
..........    530000  finished     117330  clusters
..........    540000  finished     119086  clusters
.........
   549155  finished     119563  clusters

Apprixmated maximum memory consumption: 666M
writing new database
writing clustering information
program completed !

Total CPU time 148109.70
root@BioLinuxLab:/home/bruno/Documentos/Ferramentas de Clusterização/cdhit/cd-hi
t-v4.6.1-2012-08-27# █
```

FIGURA 11. RESULTADOS GERADOS PELA EXECUÇÂO DA FERRAMENTA CD-HIT

Teste com CD-HIT (versão 4.6.1) realizado em 27/08/2015 em 549.155 sequências de proteínas da base UniPortKB-Swiss-Prot totalizando 4.11 horas.



```
>Cluster 74476
0   48aa, >sp|P17828|FIMBA_DIC... at 93.75%
1   131aa, >sp|P17830|FIMBC_DIC... at 96.95%
2   48aa, >sp|P17831|FIMBE_DIC... at 100.00%
3   48aa, >sp|P17832|FIMBF_DIC... at 95.83%
4   48aa, >sp|P17833|FIMBG_DIC... at 93.75%
5   257aa, >sp|P17834|FIMBI_DIC... *
6   257aa, >sp|P27905|FIMBX_DIC... at 95.33%
>Cluster 74477
0   257aa, >sp|Q89AZ0|FLIR_BUCB... *
>Cluster 74478
0   241aa, >sp|P02702|FOLR1_BOV... at 77.59%
1   257aa, >sp|P15328|FOLR1_HUM... *
```

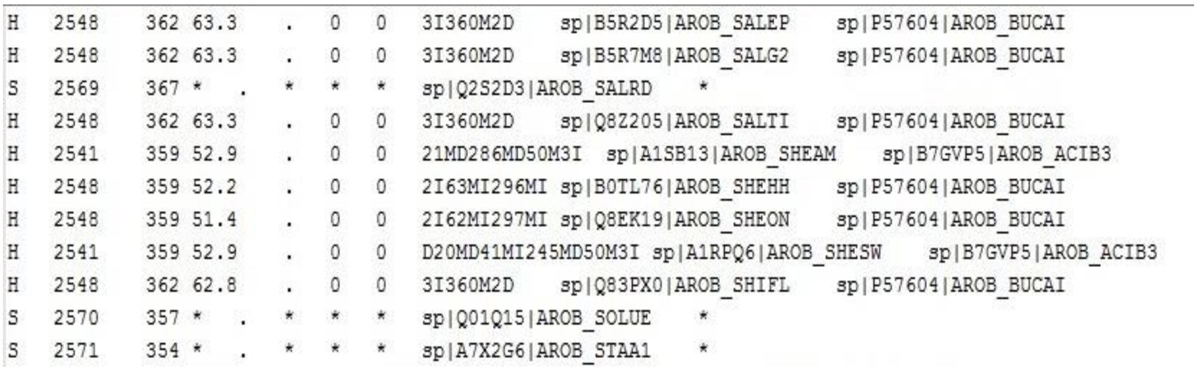
FIGURA 12. FORMATO TABULAR DE SAÍDA GERADO PELO ALGORITMO CD-HIT

Arquivo de saida do cluster utilizando o algoritmo CD-HIT com banco swiss-prot com 0.5 de identidade onde: um ">" começa um novo cluster, "*" no final significa que esta seqüência é o representante do aglomerado e "%" é a identidade entre esta sequência e o representante.

### 6.1.2 Resultados do algoritmo UCLUST

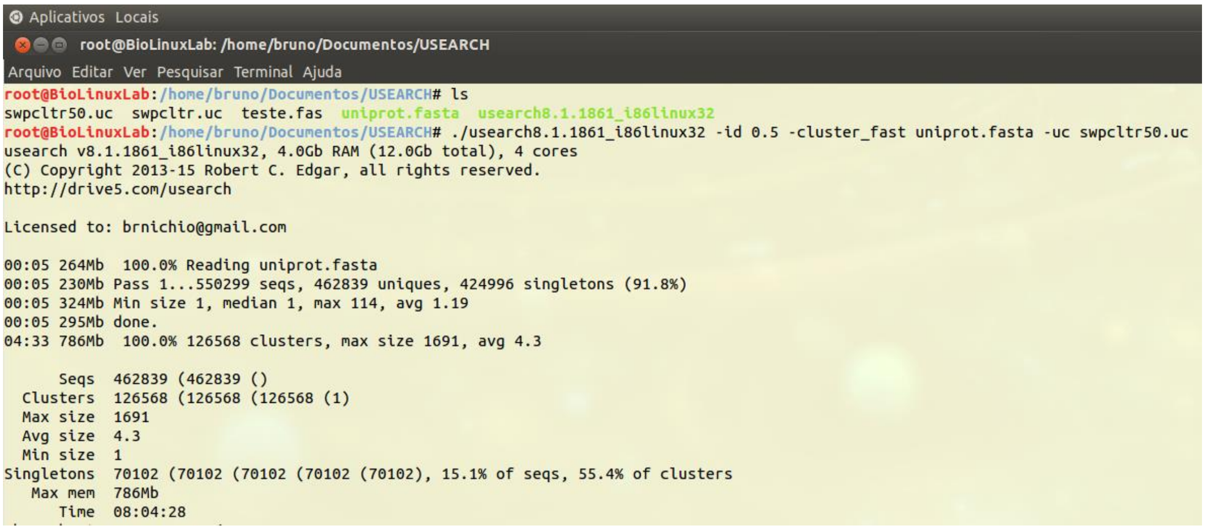
USEARCH cluster format (UC) is a tab-separated text file. UC output is supported by clustering and database search. By convention, the .uc filename extension is used. Each line is either a comment (starts with #) or a record. Every input sequence generates one record (H, S or N); additional record types give information about *clusters*. If an input sequence matched a target sequence, then the alignment and the identity computed from that alignment are also provided. Fields that do not apply to a given record type are filled with an asterisk placeholder (*).

By default, only the top hit is written to the UC output file. This reflects that the format is primarily designed for clustering, in which case -maxaccepts > 1 is used to increase cluster quality by finding closer centroid sequences. The -uc_allhits option can be used to specify that all hits are to be written (mostly useful for database searches). The -uc_allhits option is supported in version 6.0.217 and later.



```
H   2548   362 63.3   .   0   0   3I360M2D    sp|B5R2D5|AROB_SALEP    sp|P57604|AROB_BUCAI
H   2548   362 63.3   .   0   0   3I360M2D    sp|B5R7M8|AROB_SALG2    sp|P57604|AROB_BUCAI
S   2569   367 *   .   *   *   *   sp|Q2S2D3|AROB_SALRD    *
H   2548   362 63.3   .   0   0   3I360M2D    sp|Q8Z205|AROB_SALTI    sp|P57604|AROB_BUCAI
H   2541   359 52.9   .   0   0   21MD286MD50M3I  sp|A1SB13|AROB_SHEAM    sp|B7GVP5|AROB_ACIB3
H   2548   359 52.2   .   0   0   2I63MI296MI sp|B0TL76|AROB_SHEHH    sp|P57604|AROB_BUCAI
H   2548   359 51.4   .   0   0   2I62MI297MI sp|Q8EK19|AROB_SHEON    sp|P57604|AROB_BUCAI
H   2541   359 52.9   .   0   0   D20MD41MI245MD50M3I sp|A1RPQ6|AROB_SHESW    sp|B7GVP5|AROB_ACIB3
H   2548   362 62.8   .   0   0   3I360M2D    sp|Q83PX0|AROB_SHIFL    sp|P57604|AROB_BUCAI
S   2570   357 *   .   *   *   *   sp|Q01Q15|AROB_SOLUE    *
S   2571   354 *   .   *   *   *   sp|A7X2G6|AROB_STAA1    *
```

FIGURA 13: FORMATO TABULAR DE SAÌDA DO UCLUST OBTIDOS PELO ALGORITMO *CLUSTER_FAST*



Aplicativos Locais
root@BioLinuxLab: /home/bruno/Documentos/USEARCH
Arquivo Editar Ver Pesquisar Terminal Ajuda

```
root@BioLinuxLab:/home/bruno/Documentos/USEARCH# ls
swpcltr50.uc  swpcltr.uc  teste.fas  uniprot.fasta  usearch8.1.1861_i86linux32
root@BioLinuxLab:/home/bruno/Documentos/USEARCH# ./usearch8.1.1861_i86linux32 -id 0.5 -cluster_fast uniprot.fasta -uc swpcltr50.uc
usearch v8.1.1861_i86linux32, 4.0Gb RAM (12.0Gb total), 4 cores
(C) Copyright 2013-15 Robert C. Edgar, all rights reserved.
http://drive5.com/usearch

Licensed to: brnichio@gmail.com

00:05 264Mb  100.0% Reading uniprot.fasta
00:05 230Mb Pass 1...550299 seqs, 462839 uniques, 424996 singletons (91.8%)
00:05 324Mb Min size 1, median 1, max 114, avg 1.19
00:05 295Mb done.
04:33 786Mb  100.0% 126568 clusters, max size 1691, avg 4.3

      Seqs  462839 (462839 ()
  Clusters  126568 (126568 (126568 (1)
  Max size  1691
  Avg size  4.3
  Min size  1
Singletons  70102 (70102 (70102 (70102 (70102), 15.1% of seqs, 55.4% of clusters
   Max mem  786Mb
      Time  08:04:28
```

FIGURA 14. RESULTADO DO ALGORITMO UCLUST VIA TERMINAL

Resultado gerado pelo algoritmo –*cluster_fast* (versão UCLUST 8.1.1861) aplicado ao BD Swiss-Prot gerando 126.568 Cluster totais, 70.102 *Clusters* únicos em um tempo total de aproximadamente 8 horas.

### 6.1.3 Resultados do algoritmo RAFTS3groups

Conforme mencionamos anteriormente, a análise de clusterização pelo algoritmo RAFTS3groups foi feito em ambiente Matworks MATLAB (versão 2012b). O resultado do algoritmo é expresso em formato matricial ou vetores, gerando 3 informações básicas de posição das sequências clusterizadas (igrp) *clusters* totais (contall) e *clusters* representativos (contg2) ilustrado na FIGURA X.

TABELA 3. CLUSTERIZAÇÃO EM DIFERENTES *SELF-SCORES* POR RAFTS3GROUPS

| ***SelfScore*** | *Clusters* Únicos | *Clusters* Representativos | *Clusters* Totais | Tempo de Análise (Horas)* |
|---|---|---|---|---|
| **30** | 55.042 | 34.399 | 89.441 | 6,3 |
| **40** | 64.082 | 39.759 | 103.841 | 9,6 |
| **50** | 77.357 | 46.848 | 124.205 | 10,5 |
| **60** | 97.536 | 55.268 | 152.804 | 11,1 |
| **70** | 126.840 | 63.551 | 190.391 | 11,5 |
| **80** | 168.754 | 69.610 | 238.364 | 12,7 |
| **90** | 231.573 | 70.589 | 302.162 | 13,6 |

Número de *clusters* obtidos em diferentes *self-scores* pelo algoritmo RAFTS3groups em ambiente Matworks MATLAB v2012b no Sistema Operacional BioLinux 8.

## 6.2 Resultados de análise entre CD-HIT, UCLUST e RAFTS3groups

TABELA 4. NÚMERO DE SEQUÊNCIAS POR *CLUSTER* OBTIDOS POR CD-HIT, UCLUS E RAFTS3GROUPS

| | CD-HIT | UCLUST | RAFTS3groups |
|---|---|---|---|
| *Cluster 1* | 8 | 7 | 7 |
| *Cluster 2* | 5 | 5 | 6 |
| *Cluster 3* | 5 | 5 | 5 |
| *Cluster 4* | 4 | 4 | 4 |
| *Cluster 5* | 4 | 2 | 3 |
| *Cluster 6* | 3 | 2 | 3 |
| *Cluster 7* | 5 | 2 | 3 |
| *Cluster 8* | 3 | 2 | 3 |
| *Cluster 9* | 2 | 2 | 2 |
| *Cluster 10* | 2 | 2 | 2 |

| | | | |
|---|---|---|---|
| *Cluster 11* | 4 | 3 | 2 |
| *Cluster 12* | 7 | 2 | 2 |
| *Cluster 13* | 2 | 2 | 2 |
| *Cluster 14* | 2 | 2 | 2 |
| *Cluster 15* | 2 | 2 | 2 |
| *Cluster 16* | 25 | 16 | 27 |
| *Cluster 17* | 13 | 10 | 13 |
| *Cluster 18* | 6 | 7 | 9 |
| *Cluster 19* | 7 | 2 | 7 |
| *Cluster 20* | 9 | 2 | 6 |
| *Cluster 21* | 6 | 5 | 5 |
| *Cluster 22* | 4 | 4 | 4 |
| *Cluster 23* | 5 | 2 | 4 |
| *Cluster 24* | 6 | 3 | 3 |
| *Cluster 25* | 3 | 3 | 3 |
| *Cluster 26* | 3 | 10 | 3 |
| *Cluster 27* | 3 | 3 | 3 |
| *Cluster 28* | 2 | 2 | 2 |
| *Cluster 29* | 2 | 2 | 2 |
| *Cluster 30* | 2 | 2 | 2 |
| *Cluster 31* | 2 | 2 | 2 |
| *Cluster 32* | 2 | 2 | 2 |
| *Cluster 33* | 2 | 2 | 2 |
| *Cluster 34* | 67 | 49 | 56 |
| *Cluster 35* | 13 | 29 | 37 |
| *Cluster 36* | 8 | 2 | 23 |
| *Cluster 37* | 55 | 97 | 167 |
| *Cluster 38* | 7 | 2 | 14 |
| *Cluster 39* | 13 | 12 | 12 |
| *Cluster 40* | 19 | 10 | 10 |
| *Cluster 41* | 8 | 8 | 9 |
| *Cluster 42* | 10 | 8 | 8 |
| *Cluster 43* | 8 | 5 | 7 |

| | | | |
|---|---|---|---|
| *Cluster 44* | 7 | 7 | 7 |
| *Cluster 45* | 6 | 6 | 6 |
| *Cluster 46* | 6 | 6 | 6 |
| *Cluster 47* | 5 | 5 | 5 |
| *Cluster 48* | 5 | 5 | 5 |
| *Cluster 49* | 6 | 4 | 5 |
| *Cluster 50* | 5 | 5 | 5 |

Resultados obtidos selecionando 50 sequências representativas de 50 *clusters* correspondentes entre as ferramentas CD-HIT, UCLUST e RAFTS3groups.

# REFERÊNCIAS

ALTHENHOFF, A. M.; & DESSIMOZ, C. (2009). Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods. *PLoS Computational Biology*, *5*(1), e1000262.

APWEILER, R., BAIROCH, A., WU, C. H., BARKER, W. C., BOECKMANN, B., FERRO, S., … Yeh, L.-S. L. (2004). UniProt: the Universal Protein knowledgebase. *Nucleic Acids Research*, *32*(Database issue), D115–9.

BINNEWIES, T. T.; MOTRO, Y.; HALLIN, P. F.; et al. Ten years of bacterial genome sequencing: comparative-genomics-based discoveries. Functional & Integrative Genomics, v. 6, n. 3, p. 165–185, 2006.

BITARD-FEILDEL, T.; KEMENA, C.; GREENWOOD, J. M.; & BORNBERG-BAUER, E. (2015). Domain similarity based orthology detection. *BMC Bioinformatics*, *16*(1), 154.

CARDOSO, R. L. A. (2015). Análise genômica comparativa de bactérias do gênero *Herbaspirillum*.

CHEN, F., MACKEY, A. J., VERMUNT, J. K., & ROOS, D. S. (2007). Assessing Performance of Orthology Detection Strategies Applied to Eukaryotic Genomes. *PLoS ONE*, *2*(4), e383.

CHEN, T., WU, T. H., NG, W. V, & LIN, W. (2010). DODO: an efficient orthologous genes assignment tool based on domain architectures. Domain based ortholog detection. *BMC Bioinformatics*, *11 Suppl 7*(Suppl 7), S6. http://doi.org/10.1186/1471-2105-11-S7-S6

CURTIS, D. S., PHILLIPS, A. R., CALLISTER, S. J., CONLAN, S., & MCCUE, L. A. (2013). SPOCS: software for predicting and visualizing orthology/paralogy relationships among genomes. *Bioinformatics*, *29*(20), 2641–2642.

DALQUEN, D. A., ALTENHOFF, A. M. , GONNET, G. H. and DESSIMOZ, C. (2013) The impact of gene duplication, insertion, deletion, lateral gene transfer and sequencing error on orthology inference: a simulation study. PloS One, 8, e56925.

EDGAR, R.C. (2010) Search and clustering orders of magnitude faster than BLAST, *Bioinformatics* 26(19), 2460-2461.

EMMS, D. M., & KELLY, S. (2015). OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biology*, *16*(1), 157.

FITCH W. M. (2000) Homology a personal view on some of the problems. Trends Genet 16: 227–31

FREEMAN, S. & HERRON, J., C. *Análise Evolutiva* - 4ª Edição: Porto Alegre: ArtMed Editora, 2009

HUANG, Y., Niu, B., Gao, Y., Fu, L., & Li, W. (2010). CD-HIT Suite: A web server for clustering and comparing biological sequences. *Bioinformatics*, *26*(5), 680–682.

JENSEN RA. Orthologs and paralogs - we need to get it right. *Genome Biology*. 2001;2(8): interactions1002.1-interactions1002.3

KIM, K., KIM, W., & KIM, S. (2011). ReMark: An automatic program for clustering orthologs flexibly combining a Recursive and a Markov clustering algorithms. *Bioinformatics*, *27*(12), 1731–1733.

KOONIN E.V. "Orthologs, paralogs, and evolutionary genomics". *Annual Review of Genetics* 39: 309–38, 2005

KRISTENSEN, D. M., WOLF, Y. I., MUSHEGIAN, A. R., & KOONIN, E. V. (2011). Computational methods for Gene Orthology inference. *Briefings in Bioinformatics*, *12*(5), 379–391. http://doi.org/10.1093/bib/bbr030

KUZNIAR, A., VAN HAM, R. C. H. J., PONGOR, S., & LEUNISSEN, J. A. M. (2008). The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics*, *24*(11), 539–551.

LECHNER, M.; FINDEIB, S., STEINER, L., MARZ, M., STADLER, P. F., & PROHASKA, S. J. (2011). Proteinortho: Detection of (Co-)orthologs in large-scale analysis. *BMC Bioinformatics*, *12*(1), 124.

LI, W., & Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, *22*(13), 1658–1659.

LINARD, B. (2011). OrthoInspector: comprehensive orthology analysis and visual exploration. *BMC Bioinformatics*, *12*(11), 1471.

LUKJANCENKO, O.; USSERY, D. W.; WASSENAAR, T. M. Comparative Genomics of *Bifidobacterium*, *Lactobacillus* and Related Probiotic Genera. Microbial Ecology, v. 63, n. 3, p. 651–673, 2012.

MAFRA, T.; Bioinformaticando – Praticando um pouco de Bioinformática. Disponível em: http://bioinformaticando.blogspot.com.br/2012/04/orthomcl-busca-por-genes-ortologos.html?view=sidebar, 2012 Acesso em: 03/05/2016.

COIMBRA, N. A. R. Metodologia Computacional para o Estudo de Genes com vizinhança conectada: Análise do cluster nif genômica. Dissertação (Mestrado em Bioinformática) – Setor de Educação Profissional e Técnológica, Universidade Federal do Paraná, Curitiba, 2015.

PROSDOCIMI F., Curso Online de Introdução à Bioinformática, 2007 disponível em: http://www2.bioqmed.ufrj.br/prosdocimi/FProsdocimi07_CursoBioinfo.pdf.

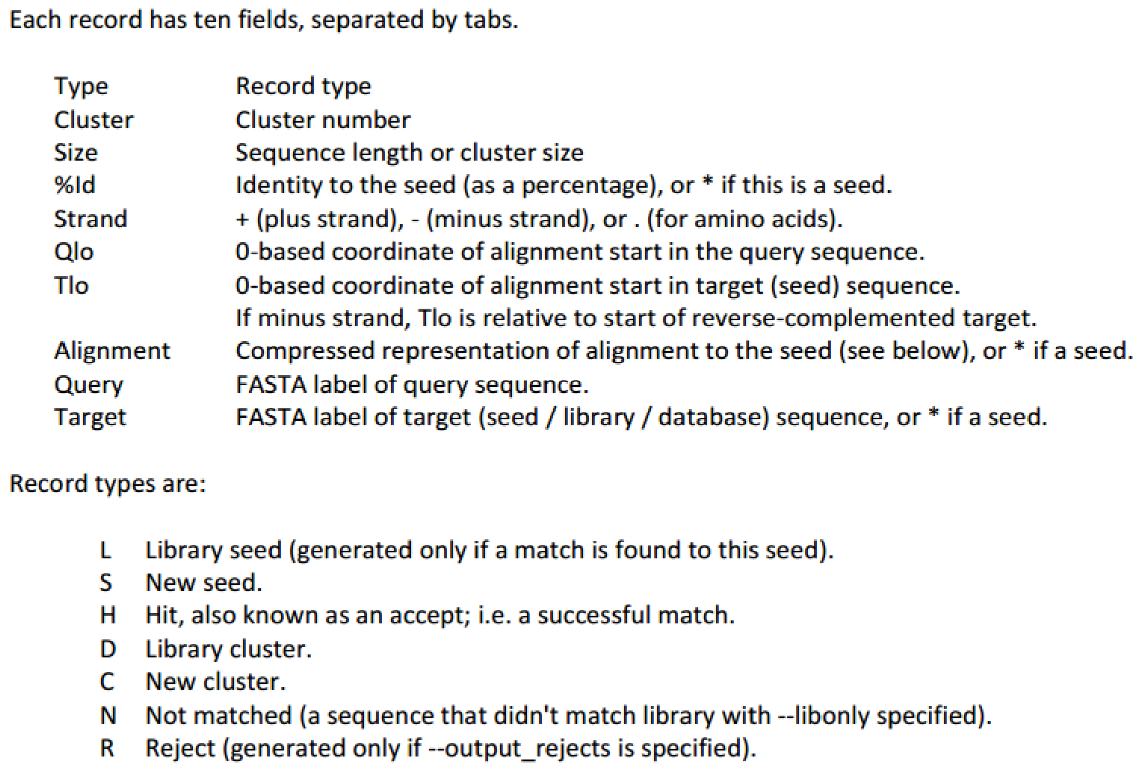
VIALLE, A. R. SILA – Ferramenta de alto desempenho para anotação automática genômica. Dissertação (Mestrado em Bioinformática) – Setor de Educação Profissional e Técnológica, Universidade Federal do Paraná, Curitiba, 2013.

WAGNER, I.; VOLKMER, M.; SHARAN, M.; VILLAVECES, J. M.; OSWALD, F.; SURENDRANATH, V.; & HABERMANN, B. H. (2014). morFeus: a web-based program to detect remotely conserved orthologs using symmetrical best hits and orthology network scoring. *BMC Bioinformatics*, *15*(1), 263.

WANG, Y.; COLEMAN-DERR, D.; CHEN, G.; & GU, Y. Q. (2015). OrthoVenn: a web server for genome wide comparison and annotation of orthologous *clusters* across multiple species. *Nucleic Acids Research*, *43*(W1), W78–W84.

ZAHA, A.; FERREIRA, H. B.; PASSAGLIA, L. M. P. (Org.). Biologia molecular básica. 5. ed. Porto Alegre: Artmed, 2014. 416 p.

# ANEXO 1 – UCLUST USERGUIDE



Each record has ten fields, separated by tabs.

| | |
|---|---|
| Type | Record type |
| Cluster | Cluster number |
| Size | Sequence length or cluster size |
| %Id | Identity to the seed (as a percentage), or * if this is a seed. |
| Strand | + (plus strand), - (minus strand), or . (for amino acids). |
| Qlo | 0-based coordinate of alignment start in the query sequence. |
| Tlo | 0-based coordinate of alignment start in target (seed) sequence.<br>If minus strand, Tlo is relative to start of reverse-complemented target. |
| Alignment | Compressed representation of alignment to the seed (see below), or * if a seed. |
| Query | FASTA label of query sequence. |
| Target | FASTA label of target (seed / library / database) sequence, or * if a seed. |

Record types are:

| | |
|---|---|
| L | Library seed (generated only if a match is found to this seed). |
| S | New seed. |
| H | Hit, also known as an accept; i.e. a successful match. |
| D | Library cluster. |
| C | New cluster. |
| N | Not matched (a sequence that didn't match library with --libonly specified). |
| R | Reject (generated only if --output_rejects is specified). |

FIGURA 15: *UCLUST USERDGUIDE - TABLE OUTPUT FORMAT LEGEND AND FIELDS*

Disponibility: http://www.drive5.com/uclust/uclust_userguide_1_1_579.pdf