sits.rep: Pesquisa Reprodutível em Classificações de Uso e Cobertura da Terra

Rafael M. Mariano[1], **Gilberto R. Queiroz**[1], **Pedro R. Andrade**[1], **Rafael Santos**[1]

[1]Instituto Nacional de Pesquisas Espaciais (INPE)
Av. dos Astronautas, 1758 – 12227-010 – São José dos Campos – SP – Brasil

{rafael.mariano, gilberto.queiroz, pedro.andrade, rafael.santos}@inpe.br

Resumo. *A reprodutibilidade de pesquisas tem sido um tópico de grande discussão na comunidade científica. Esta questão tem motivado os principais periódicos a elaborarem documentos de boas práticas que ajudam os pesquisadores a organizarem os dados, códigos e artefatos de suas publicações para assegurar a reprodução dos trabalhos. Por causa disto, diversas ferramentas computacionais têm sido desenvolvidas com o objetivo de lidar com as questões de reprodutibilidade científica. Este trabalho apresenta uma ferramenta tecnológica para se obter reprodutibilidade de experimentos científicos realizados na criação de mapas de uso e cobertura da terra baseadas em técnicas de aprendizado de máquina com o pacote R denominado* `sits` *(Satellite Image Time Series). Esta ferramenta, denominada* `sits.rep`*, auxilia pesquisadores em todos os passos de seus experimentos, aumentando a produtividade das equipes que desenvolvem códigos de classificações de uso e cobertura da terra, uma vez que os pesquisadores podem se dedicar exclusivamente em produzir melhores classificações.*

Abstract. *The reproducibility of research has been a topic of great discussion in the scientific community. This question has motivated international journals to create good practices documents that help researchers to organize data, source codes, and artifacts of their publications to ensure the reproduction of publications. Consequently, several computational tools have been developed with the aim of dealing with issues of scientific reproducibility. This paper presents a tool to obtain reproducibility of scientific experiments in the context of land use and land cover classifications based on machine learning techniques with the R package called* `sits` *(Satellite Image Time Series). This tool, called* `sits.rep`*, assists researchers in all the steps of their experiments, thus increasing the productivity of the teams that develop land use classifications, once they can focus exclusively on producing better classifications.*

1. Introdução

A reprodutibilidade é um dos pilares do método científico, tendo como propósito garantir que os resultados observados nos experimentos originais possam ser reproduzidos de forma independente [Peng 2011, Vasilevsky et al. 2013, McNutt 2014]. Este princípio permite que outros pesquisadores possam avaliar as premissas de uma metodologia proposta, podendo-se descobrir limitações ou até falhas, a partir das quais novos trabalhos podem ser desenvolvidos, avançando assim o conhecimento científico. Entretanto, alguns estudos apontam a existência de pesquisas científicas, publicadas após

revisão por pares, com informações incompletas para a reprodutibilidade das mesmas [Ioannidis 2005, Prinz et al. 2011, Baker 2016, Gertler et al. 2018]. Estes estudos alegam falta de rigor técnico e transparência sobre os dados e métodos usados durante o desenvolvimento dos experimentos.

Para lidar com esses problemas, a comunidade científica tem desenvolvido documentos de boas práticas[1], *frameworks* e ferramentas computacionais para auxiliar os pesquisadores a organizarem e compartilharem dados, códigos e demais artefatos de suas publicações [Gentleman and Lang 2007, Nosek et al. 2015, Gil et al. 2016, Beaulieu-Jones and Greene 2017]. Alguns periódicos já obrigam os autores a depositarem dados e códigos em algum repositório público, de forma a reforçar a importância destas informações para possibilitar a reprodutibilidade das pesquisas.

No que diz respeito às ferramentas computacionais, existem soluções genéricas que tratam as questões de reprodutibilidade de maneira posterior à realização de todas as fases da pesquisa [Chirigati et al. 2013, Landau 2018, Nüst et al. 2017, Govoni et al. 2019]. Contudo, os processos envolvidos em uma pesquisa científica não são lineares, envolvendo diversas fases de experimentação que podem explorar diferentes abordagens e parâmetros até que seja encontrado algum resultado interessante, que então é usado para publicação. Se questões relativas à reprodutibilidade não forem consideradas durante a realização da pesquisa, dificilmente o resultado final publicado poderá ser reproduzido devido às dificuldades para preparação do ambiente, execução dos experimentos, ou até para se encontrar a versão do código usada para produzir os resultados originais. Assim, é de suma importância tratar a reprodutibilidade durante o desenvolvimento da pesquisa, e não somente ao final do processo.

Existem ferramentas computacionais que buscam tratar das questões de reprodutibilidade durante a fase de experimentação [Di Tommaso et al. 2017, Greff et al. 2017]. Estas ferramentas atendem às necessidades específicas de cada área de pesquisa, tais como diferentes formatos de dados, linguagens de programação específicas e o uso ou não de algoritmos estocásticos. Entretanto, estas ferramentas acabam impondo ao pesquisador um trabalho extra por não serem integradas ao ambiente de experimentação. Com base nisso, uma ferramenta para fins específicos e bem definidos, integrada ao ambiente de experimentação, pode facilitar o processo ligado às questões de reprodutibilidade, coletando todas as informações necessárias durante a execução das etapas de uma pesquisa.

Este trabalho investiga as questões de reprodutibilidade em pesquisas relacionadas ao mapeamento automatizado de classificações de uso e cobertura da terra. Em geral, estas pesquisas envolvem grandes volumes de dados e equipes multidisciplinares, bem como atividades com grande complexidade, não existindo qualquer ferramenta que auxilie na reprodutibilidade dos experimentos. Este trabalho tem como objetivo apresentar uma ferramenta computacional para organizar e armazenar dados, códigos e demais informações relevantes produzidas durante o desenvolvimento desse tipo de pesquisa, com base no pacote R denominado *Satellite Image Time Series* (`sits`[2]) [Camara et al. 2018]. A partir da coleta sistematizada destas informações, a ferramenta automatiza o processo de reprodutibilidade dos experimentos desenvolvidos usando o pacote `sits`.

[1] Por exemplo, `https://www.nature.com/sdata/policies/data-policies` e `https://www.sciencemag.org/authors/science-journals-editorial-policies`.

[2] `https://github.com/e-sensing/sits`

2. Satellite Image Time Series - sits

Na área de Observação da Terra são desenvolvidos diversos trabalhos com o intuito de monitorar e avaliar o ambiente natural e as mudanças causadas pelo homem [Group on Earth Observations 2020]. Dados podem ser obtidos através de sensores presentes nos satélites, que obtêm informações espectrais relativas à cobertura terrestre. Esses dados, disponíveis na forma de imagens, são usados pelos cientistas em suas pesquisas para analisar o uso e cobertura da terra. A disponibilização aberta das imagens de sensoriamento remoto tem crescido significativamente, principalmente através da adoção de políticas de acesso aberto e livre por instituições de pesquisa [Soille et al. 2018]. Diante desse grande volume de dados, os pesquisadores enfrentam o desafio de projetar tecnologias que possam aproveitar todo o potencial desse conjunto de dados para realizar análises e auxiliar nas políticas públicas [Camara et al. 2016].

Uma iniciativa aberta para classificação de uso e cobertura da terra utilizando séries temporais de imagens de satélites é o `sits` [Camara et al. 2018]. O `sits` é um pacote desenvolvido na linguagem R que fornece ferramentas para visualização, suavização, clusterização e classificação de séries temporais. Para as classificações, são utilizadas amostras de dados *in situ*, juntamente com as respectivas séries temporais destas amostras e algoritmos de aprendizagem de máquina, como o *Support Vector Machine* (SVM) e *Multilayer Perceptron* (MLP). O grupo de pesquisa responsável pelo desenvolvimento do `sits` armazena todas as amostras de treinamento relevantes no pacote `inSitu`[3]. Da mesma forma, todos os dados de séries temporais usados para a classificação estão disponíveis através do pacote `EOCubes`[4].

O conjunto de classes definidas para as amostras será usado para produzir a legenda final da classificação. Para cada pixel, é produzida uma probabilidade dele pertencer a cada uma das classes disponíveis, em cada intervalo de tempo. Normalmente, a classe com maior probabilidade é escolhida como sendo a classificação do pixel. Ao final do processo, mapas temporais de uso e cobertura da terra são produzidos para a região estudada.

Após as classificações, podem ser realizados pós-processamentos, incluindo a aplicação de máscaras de classes não consideradas na classificação, filtros para suavização das classificações, bem como regras de transições para remover inconsistências temporais. Esse processo de pós-processamento é experimental, podendo envolver a aplicação de diferentes passos sobre um mesmo conjunto de dados, de forma a entender ou melhorar os resultados das classificações. Desta forma, a criação de informações após as classificações segue um fluxo que pode conter bifurcações. Ao final deste processo, apenas um destes caminhos será escolhido como sendo a classificação final. A Figura 1 ilustra esse fluxo experimental do pós-processamento utilizando o pacote `sits.validate`, que é usado para pós-processar e validar as classificações, bem como agregar as saídas em diferentes representações.

Apesar do pacote `sits` ser completo, é necessário um esforço significativo por parte dos pesquisadores para garantir as questões de reprodutibilidade, incluindo cuidados relativos à estocasticidade dos algoritmos de aprendizagem de máquina, versionamento

[3] `https://github.com/e-sensing/inSitu`
[4] `https://github.com/e-sensing/EOCubes`

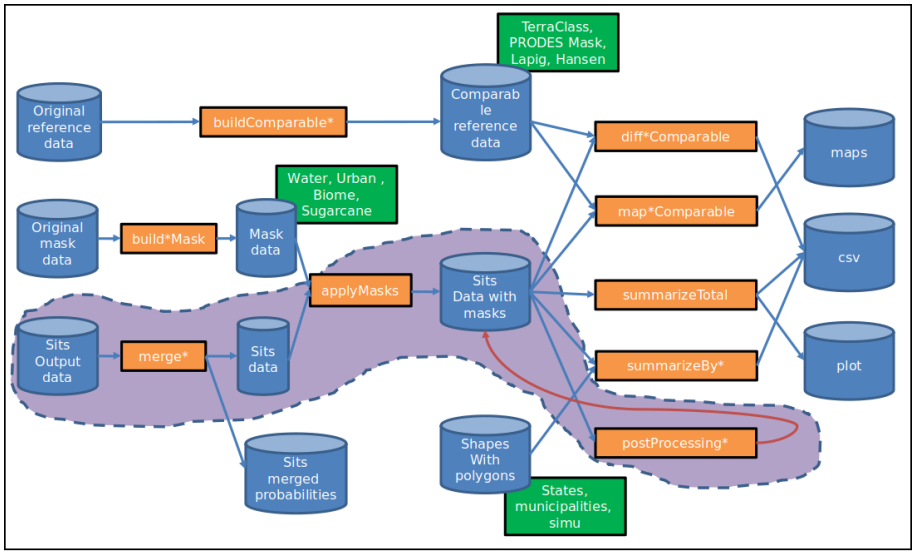

Figura 1. Processo de pós-processamento e validação das classificações do sits.

de dados e software, dependências de outros pacotes e até a identificação do *script* usado na geração de uma determinada classificação. Além disso, as pesquisas desenvolvidas com este pacote envolvem grupos multidisciplinares, que executam diversos experimentos, com grande interação da equipe. Dessa forma, um experimento desenvolvido por um pesquisador poderá ser usado como entrada para um novo experimento feito por um segundo pesquisador, e assim por diante, criando-se uma árvore de experimentos. Como o registro de todas as informações deve ser feito manualmente, recuperá-lo após o final do processo de classificação e pós-processamento se mostra uma tarefa difícil, podendo até inviabilizar a disponibilização de todos os elementos necessários à reprodutibilidade do trabalho.

3. Problema

A questão de se garantir a reprodutibilidade é inerente ao processo de desenvolvimento da ciência. Em busca de se obter contribuições inovadoras, pesquisadores costumam alterar com frequência os experimentos e executá-los até que sejam encontrados resultados significativos. Registrar todos os passos executados até se obter o resultado final requer um esforço de coleta de informações que serão descartadas, com exceção dos experimentos que obtiveram os melhores resultados. Em experimentos *in silico*, atitudes simples, como copiar o código usado para o diretório com os resultados, normalmente são ignoradas.

As ferramentas encontradas tanto na academia quanto na indústria buscam gerenciar a organização dos dados e metadados produzidos pelos experimentos durante ou depois do desenvolvimento da pesquisa. Foram investigadas diferentes formas de aplicar estas ferramentas no contexto do pacote `sits`. Algumas dessas ferramentas não possuem suporte à linguagem R, nativa do pacote `sits`, enquanto outras não permitem tratar problemas específicos de linguagens ou pacotes, armazenando somente os dados de entrada, saída e o *script* executado. Além disso, existem casos onde o pesquisador é responsável por coletar todas as informações necessárias para se garantir a reprodutibilidade, tais como semente aleatória, versões e dependências do ambiente onde os códigos são executados, assim como é necessário para pesquisadores que usam o `sits`.

Quando existem perdas de informações relativas ao processo usado para produzir uma determinada classificação de uso e cobertura da terra, não é possível reproduzir um experimento com exatidão, o que inviabiliza todo e qualquer esforço executado na cadeia de experimentos posterior a ele. Esses problemas podem ter um impacto grande no processo de publicação das pesquisas científicas que possuem resultados gerados a partir desse pacote. Desta forma, é altamente recomendável ter uma ferramenta que esteja presente desde o inicio do desenvolvimento da pesquisa e que seja capaz de armazenar todos os códigos, dados de entrada, dados de saída e informações de ambiente de desenvolvimento, necessários para reproduzir um experimento. Este trabalho apresenta uma ferramenta para gerenciar todo o processo de desenvolvimento dos experimentos executados com o pacote `sits`, a fim de automatizar a sua reprodutibilidade. Esta ferramenta, denominada `sits.rep`, será apresentada a seguir.

4. sits.rep

A ferramenta `sits.rep`[5] é desenvolvida na linguagem R com o propósito de coletar informações durante a execução de experimentos realizados com o pacote `sits`. Esta coleta de informações permite criar um ambiente baseado em tecnologia de contêineres para reprodução dos experimentos. Ela possui um conjunto de funções para executar classificações de uso e cobertura da terra e permitir o pós-processamento sobre os resultados dessas classificações, mantendo o encadeamento entre esses processos. Essa forma de lidar com os experimentos possibilita o pesquisador reproduzir qualquer etapa da pesquisa sem exigir um trabalho extra para gerenciar todo o ambiente utilizado para o experimento. O `sits.rep` é orientado a duas atividades principais na criação dos mapas de uso e cobertura: criação dos experimentos e reprodução dos resultados.

4.1. Criação dos Experimentos

Para a criação dos mapas, os pesquisadores inicialmente desenvolvem e executam *scripts* em R usando o pacote `sits` para classificar um conjunto de séries temporais de imagens de satélite. Posteriormente, podem ser realizadas ações de pós-processamento sobre os resultados da classificação com o propósito de melhorar a acurácia dos resultados. Durante a execução desses *scripts*, o `sits.rep` coleta metadados referentes ao contexto das suas execuções para identificar quais as dependências entre os *scripts* e assim poder assegurar que os mesmos sejam organizados em um experimento reprodutível.

As dependências entre os *scripts* são estruturadas na forma de uma *árvore*, onde o nó raiz se refere a um *script* de classificação e os demais, aos *scripts* de pós-processamento. A Figura 2 exemplifica esse conceito com o *script Classificação* inicializando uma árvore de processos e produzindo um resultado chamado de *Dados Classificados*. A partir dos *Dados Classificados*, são executados dois *scripts* de pós-processamento independentes, cujos resultados não estarão relacionados, chamados de *Classificação com Máscara de Nuvem* e *Classificação com Filtro Bayesiano*. Sobre este último nó são executados dois outros *scripts*, produzindo resultados também independentes. Nesta árvore, um *experimento* é caracterizado por um caminho a partir do nó raiz em direção às folhas, contendo um ou mais nós. Cada experimento conterá todos os metadados necessários para permitir a sua reprodução.

[5] `https://github.com/RafaMariano/sits.rep`

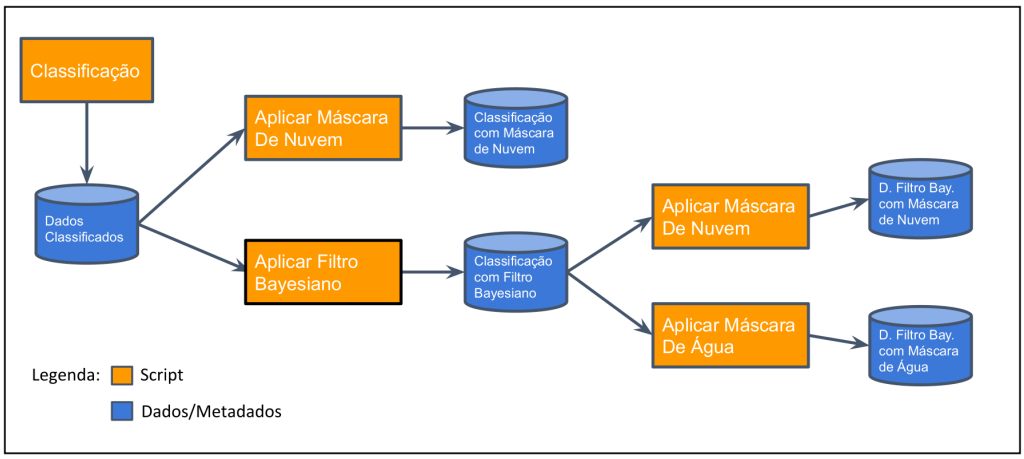

Figura 2. Relação de árvore de dependências com os processos de classificação e pós processamento.

Uma vantagem do uso do pacote `sits` para a reprodutibilidade é a disponibilização dos dados de entrada de forma livre e aberta. Os dados de amostras são todos armazenados no pacote `inSitu`. Os dados de sensoriamento remoto são disponíveis através do pacote `EOCubes`, sendo disponibilizados através de serviços Web. Dessa forma, o `sits.rep` não precisa armazenar todos os dados brutos usados durante o processamento, desde que os pacotes `inSitu` e `EOCubes` sejam devidamente usados. Este trabalho assume que os dados usados para o treinamento e classificação são obtidos através destes dois pacotes.

4.2. Reprodução dos Experimentos

O processo de reprodução dos experimentos é aplicado sobre um nó da árvore de classificações a ser reproduzido. O `sits.rep` então usa todas as informações relativas ao caminho da árvore de classificações do nó raiz até o nó escolhido. Por exemplo, para reproduzir o *script* Aplicar Filtro Bayesiano, mostrado na árvore da Figura 2, o `sits.rep` usa tanto este *script* quanto o *script* de classificação. Assim, as demais etapas, como o *script* Aplicar Máscara de Nuvem, não são necessárias.

Para assegurar que o ambiente original de experimentação possa ser devidamente compartilhado com outros pesquisadores, o `sits.rep` fornece a possibilidade de preparação de um ambiente virtual de reprodução, baseado na tecnologia de contêiner Docker[6]. Esse passo é fundamental, uma vez que possibilita outros pesquisadores a verificação dos códigos e resultados obtidos em um ambiente que replica o experimento original, aumentando a confiabilidade da pesquisa. Para isso, o `sits.rep` usa um sistema de virtualização em conjunto com as informações de todo o caminho selecionado da árvore de classificações para recriar o ambiente original e um único *script* a partir de todos os *scripts* do caminho, chamado de *script reprodutível*. Se forem usados os dados dos pacotes `inSitu` e `EOCubes`, o compartilhamento do `script` reprodutível pode ser realizado de forma simples devido à necessidade de se compartilhar apenas os metadados e não os dados brutos, economizando considerável espaço de armazenamento.

[6] `https://www.docker.com/resources/what-container`

5. Resultados

Esta seção apresenta um estudo de caso que demonstra a capacidade do `sits.rep` de ajudar pesquisadores a gerar resultados reprodutíveis. O código-fonte do estudo de caso é mostrado na Figura 3. Ele inicia com a definição de uma árvore, com o nome `deep`, onde serão armazenados todos os resultados. Em seguida, o *script* de classificação é executado, como mostrado na linha 3. Este *script* `classification.R` utiliza os dados de amostras do pacote `inSitu` para treinar um modelo de aprendizagem de máquina com o algoritmo de *Deep Learning*[7]. Após o treinamento, são solicitadas ao pacote `EOCubes` as imagens de satélite do produto `MOD13Q1` de uma região de estudo localizada no Estado do Mato Grosso. Estas imagens são classificadas usando o modelo treinado. O `sits.rep` usa uma estratégia de introspecção no código do pacote. Esta estratégia sobrescreve em tempo de execução um conjunto de funções previamente escolhidas para coletar automaticamente diversos metadados para a reprodutibilidade do *script* de classificação. Esses metadados, relativos a informações sobre os dados de entrada, resultados, modelo treinado, versões dos pacotes e semente da execução, serão usados posteriormente para reproduzir o experimento sob o mesmo contexto da execução original. No final da execução do *script* de classificação, os resultados serão armazenados em um diretório denominado `classification`, sob a árvore de dependência.

```
1  library(sits.rep)
2  sits.rep::useTree("deep")
3  sits.rep::classify(script = "classification.R")
4
5  sits.rep::pos_processing(parent = "classification",
6      process = "smooth", func = sits.rep::smooth)
7
8  sits.rep::pos_processing(parent = "smooth",
9      process = "mosaic", func = sits.rep::merge)
10
11 sits.rep::pos_processing(parent = "classification",
12     process = "classify_mosaic", func = sits.rep::merge)
```

Figura 3. Demonstração de uso do pacote `sits.rep` para a criação dos experimentos de classificação e pós processamento.

Após finalizada a classificação, os resultados serão usados como entrada para os experimentos de pós-processamento. O código da linha 5 executa uma função de segunda ordem para aplicar um filtro Bayesiano sobre todas as imagens classificadas. Os dois próximos comandos são executados para criar mosaicos sobre as imagens do resultado do experimento inicial, bem como do experimento 'smooth'. O pacote EOCubes divide a grade original do MODIS em grades de tamanho 10×10, sendo interessante produzir apenas uma imagem para posteriormente aplicar máscaras. Ao criar um novo experimento a partir de um nó que já possua um filho (por exemplo, o da linha 11), o `sits.rep` automaticamente cria uma bifurcação na árvore.

Finalizada a execução dos experimentos, o usuário deverá escolher um dos nós da árvore para gerar o `script` reprodutível. Este `script` envolverá todos os nós a partir

[7] O código fonte do *script* de classificação pode ser encontrado no seguinte repositório: `https://github.com/RafaMariano/sits_rep_estudo_de_caso`.

```
1 library(sits.rep)
2 sits.rep::useTree("deep")
3 sits.rep::reproduce("mosaic", "mosaic_rep")
4 sits.rep::buildContainer("mosaic_rep")
```

Figura 4. Reprodução de um experimento.

da raiz até o nó selecionado. Para isso, basta informar para a função `reproduce` o nome do experimento e do diretório aonde serão armazenado os arquivos, como demonstrado na linha 3 da Figura 4. Esse *script* reprodutível conterá todos os passos aplicados até a reprodução dos mesmos resultados do nó selecionado para reprodução. Ele é armazenado em um diretório que poderá ser carregado em plataformas abertas para que qualquer usuário possa reproduzir os experimentos executados. Adicionalmente, o `sits.rep` disponibiliza uma função chamada `buildContainer`, que recebe o diretório de um `script` reprodutível, e constrói uma imagem Docker contendo todas as dependências, nas mesmas versões usadas para gerar a classificação, juntamente com o `script` reprodutível. Com isto, todo o processo de reprodutibilidade é garantido, desde a coleta das informações enquanto o usuário executa `scripts` de classificação e pós-processamento, até a geração de um contêiner reprodutível, que poderá ser disponibilizado em um repositório na Web.

6. Conclusões e Trabalhos Futuros

Questões relativas à reprodutibilidade têm sido discutidas nas diversas áreas da ciência, levando à criação de ferramentas computacionais para auxiliar os pesquisadores a organizarem os artefatos de suas publicações e assim assegurar uma maior facilidade para se garantir a reprodução de pesquisas científicas. Neste contexto, o pacote `sits.rep` sistematiza a organização e armazenamento dos dados e códigos produzidos por estudos de classificações de uso e cobertura da terra.

Através do `sits.rep`, é possível automatizar a criação de experimentos reprodutíveis e compartilhá-los com outros pesquisadores. Devido à sobrescrita das funções usadas pelo *script* de classificação, é possível reaproveitar códigos e torná-los reprodutíveis sem qualquer alteração. A partir da consolidação do `sits.rep`, acreditamos que a produtividade das equipes que desenvolvem códigos de classificações de uso e cobertura da terra aumente substancialmente, deixando os pesquisadores despreocupados com as questões de reprodutibilidade, podendo focar exclusivamente em produzir melhores classificações.

Nos trabalhos futuros temos o objetivo de produzir um serviço nas nuvens capaz de permitir que diferentes pesquisadores trabalhem em uma mesma classificação. Os experimentos serão executados em um servidor escalável capaz de processar grandes volumes de dados. Adicionalmente, serão desenvolvidas funcionalidades para visualização da árvore de classificações, mostrando os nós com melhores resultados, bem como possibilitando a inclusão de novos nós de forma visual.

7. Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil, processo 130877/2018-2.

Referências

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature News*, 533(7604):452.

Beaulieu-Jones, B. K. and Greene, C. S. (2017). Reproducibility of computational workflows is automated using continuous analysis. *Nature biotechnology*, 35(4):342–346.

Camara, G., Assis, L. F., Ribeiro, G., et al. (2016). Big earth observation data analytics: Matching requirements to system architectures. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, BigSpatial '16, page 1–6, New York, NY, USA. Association for Computing Machinery.

Camara, G., Simoes, R., Andrade, P. R., et al. (2018). e-sensing/sits: Version 1.12.5.

Chirigati, F., Shasha, D., and Freire, J. (2013). Reprozip: Using provenance to support computational reproducibility. In *Proceedings of the 5th USENIX Conference on Theory and Practice of Provenance*, TaPP'13, page 1, USA. USENIX Association.

Di Tommaso, P., Chatzou, M., Floden, E. W., et al. (2017). Nextflow enables reproducible computational workflows. *Nature biotechnology*, 35(4):316–319.

Gentleman, R. and Lang, D. T. (2007). Statistical analyses and reproducible research. *Journal of Computational and Graphical Statistics*, 16(1):1–23.

Gertler, P., Galiani, S., and Romero, M. (2018). How to make replication the norm (vol 554, pg 417, 2018). *Nature*, 555(7698):580–580.

Gil, Y., David, C. H., Demir, I., et al. (2016). Toward the geoscience paper of the future: Best practices for documenting and sharing research from data to software to provenance. *Earth and Space Science*, 3(10):388–415.

Govoni, M., Munakami, M., Tanikanti, A., et al. (2019). Qresp, a tool for curating, discovering and exploring reproducible scientific papers. *Scientific data*, 6:190002.

Greff, K., Klein, A., Chovanec, M., et al. (2017). The sacred infrastructure for computational research. In *Proceedings of the 15th Python in Science Conference (SciPy 2017)*, volume 28, pages 49–56.

Group on Earth Observations (2020). Group on earth observations - FAQ - Accessed: 2020-02-20. `https://www.earthobservations.org/g_faq.html`.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLOS Medicine*, 2(8).

Landau, W. M. (2018). The drake r package: a pipeline toolkit for reproducibility and high-performance computing. *Journal of Open Source Software*, 3(21).

McNutt, M. (2014). Reproducibility. *Science*, 343(6168):229–229.

Nosek, B. A., Alter, G., Banks, G. C., et al. (2015). Promoting an open research culture. *Science*, 348(6242):1422–1425.

Nüst, D., Konkol, M., Pebesma, E., et al. (2017). Opening the publication process with executable research compendia. *D-Lib Magazine*, 23(1/2).

Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060):1226–1227.

Prinz, F., Schlange, T., and Asadullah, K. (2011). Believe it or not: how much can we rely on published data on potential drug targets? *Nature reviews Drug discovery*, 10(9):712.

Soille, P., Burger, A., Marchi], D. D., et al. (2018). A versatile data-intensive computing platform for information retrieval from big geospatial data. *Future Generation Computer Systems*, 81:30 – 40.

Vasilevsky, N. A., Brush, M. H., Paddock, H., et al. (2013). On the reproducibility of science: unique identification of research resources in the biomedical literature. *PeerJ*, 1:e148.