



Universidade Estadual de Santa Cruz
Pró-Reitoria de Pesquisa e Pós-Graduação
Programa Institucional de Iniciação Científica

RELATÓRIO FINAL

Estrelas Variáveis Pulsantes como Indicadores de Distância
de Aglomerados Estelares

Estrelas Variáveis Pulsantes como Indicadores de Distância
de Aglomerados Globulares

Bolsista: João Pedro dos Santos Rocha

Orientador: Leandro de Oliveira Kerber

Programa: ICB

Vigência de bolsa: Agosto 2019 à Julho 2020

Ilhéus, 2020.

1 Resumo

A cada dia a Astronomia de domínio temporal se torna mais acessível. O aumento no poder de processamento, o número crescente de grandes *surveys* como a *Dark Energy Survey* (DES Collaboration, 2016), e no futuro a *Legacy Survey of Space Time* (LSST) (Sánchez et al. 2020), geram novos desafios na área. Na presente investigação nos deparamos com um desses desafios: a grande quantidade de dados para reduzir e analisar oriundos de séries temporais fotométricas de aglomerados estelares obtidos em projetos próprios de observação. Por isso nesse trabalho acabamos por nos dedicar à esse problema, porém com o foco nos desafios específicos referentes à proposta do trabalho: a detecção e obtenção de curvas de luz de estrelas no ambiente de aglomerados estelares. Como produto foi construído um *software* de redução e análise de imagens astronômicas em Python, que diminuiu o tempo do processo de dias para algumas horas. A agilidade adquirida possibilitou a realização de vários testes, o que ajudou a melhorar a qualidade do resultado final, e a experiência com a implementação serviu como uma excelente oportunidade para refinar o conhecimento sobre estas técnicas computacionais.

2 Introdução

A Astronomia de domínio temporal se torna cada vez mais acessível. Os grandes *surveys* como o *Dark Energy Survey* e no futuro o *Legacy Survey of Space Time* prometem trazer novas perspectivas ao estudo dos fenômenos de variabilidade em sistemas astrofísicos.

Em Astrofísica, fenômenos de variabilidade são aqueles que fazem com que o fluxo de determinados objetos varie com o tempo. Dentre esses objetos estão as estrelas variáveis, que por sua vez, podem apresentar diversos mecanismos pelos quais seu brilho varia. Ao estudar essas estrelas é possível aprender sobre diferentes aspectos da Astrofísica Estelar, o que torna interessante de investigá-las dentro dos aglomerados estelares.

Ao estudar as variáveis dentro de aglomerados, abre-se a possibilidade de usá-las para investigar o aglomerado ou vice versa. Isso torna muito importante o estudo do conteúdo de variáveis em aglomerados pois pode fornecer uma visão mais ampla quanto a evolução estelar no sistema (e.g. Salinas 2018), ou ajudar a estimar parâmetros do aglomerado, como a distância (e.g. Kerber et al. 2018, 2019). Da mesma forma, os parâmetros

determinados para o aglomerado (idade, avermelhamento, etc) podem ser usados para estimar parâmetros individuais das variáveis (Roelens et al. 2016).

Nos anos de 2017 e 2018 foram realizados projetos pilotos que revelaram a viabilidade de estudos de estrelas variáveis de curto período (menor que 1 dia) em aglomerados estelares utilizando os telescópios Boller & Chivens (B&C, de 0,60 m) e Perkin-Elmer (PE, de 1,60 m) ambos disponíveis no Observatório do Pico dos Dias (MG).

Em consequência foram feitos novos pedidos de tempo em 2019 e 2020, para que pudessem ser ampliadas as séries temporais de imagens desses aglomerados. Com isso mais campanhas de observação foram realizadas e mais dados foram coletados, o que exige um aprimoramento na metodologia de redução e análise de imagens aplicada por nós e utilizada para a detecção de estrelas variáveis nestes objetos.

Logo nas primeiras observações ficou claro um problema que já vinha sido discutido. Devido a alta cadência necessária para estudar esse tipo de fenômeno (entre 50 e 100 imagens por hora) uma grande quantidade de dados fora gerado. Esse volume seria inviável de reduzir usando a metodologia empregada até então, pois a mesma requeria constante interação manual via o *software* padrão para isso, o IRAF¹ (*Image Reduction and Analysis Facility*). O que seria problemático por tomar muito tempo e ser passível de erros por fatores humanos.

O IRAF é historicamente o padrão utilizado para a redução e análise de imagens astronômicas. Se trata de uma coleção de programas desenvolvido no NOAO (*National Optical Astronomy Observatory*) por volta do início dos anos 80, portanto é um sistema de quase 40 anos de idade. Embora continue sendo muito relevante, está em fase de finalização de suporte.

Devido à limitações dos computadores época que foi construído, o IRAF possui certos fatores limitantes. Diversas funcionalidades presentes em sistemas mais modernos fazem falta ao se automatizar rotinas usando *scripts* do IRAF, essas restrições foram percebidas e por consequência soluções diferentes foram procuradas.

A solução adotada foi a mesma proposta pelo STScI (*Space Telescope Science Institute*): utilizar a linguagem de programação Python. No fim do século XX pesquisadores do STScI estudavam a possibilidade do uso do Python para a execução das operações no instituto (Barret, 2001). Desde então muito aconteceu, e hoje o ecossistema científico

¹<http://ast.noao.edu/data/software>

disponível na linguagem possibilitou a migração do IRAF para o Python. Por isso hoje o Python é uma excelente opção para a realização de projetos em Astronomia e vem sendo extensivamente utilizado no STScI e em diversos centros de pesquisa em Astronomia e Astrofísica no mundo.

A consolidação do Python na área da Astronomia veio com o projeto Astropy. Com o amadurecimento dos pacotes de computação científica do Python, como o Numpy, Scipy e Matplotlib, desenvolvedores começaram a criar ferramentas para Astronomia. No entanto ainda não havia sinergia entre os desenvolvedores por falta de comunicação dentro da comunidade. Então motivados por esse problema em 2011 foi iniciado o projeto Astropy (The Astropy Collaboration, 2013). O projeto organizou os esforços da comunidade, unificando e padronizando as ferramentas. Hoje o projeto já tem mais de 40 pacotes afiliados além dos 8 pacotes presentes na biblioteca padrão.

Portanto, no trabalho aqui desenvolvido, a proposta foi a de utilizar as ferramentas disponíveis até então para montar rotinas para a redução e análise automatizada de séries temporais de imagens no padrão FITS (*Flexible Image Transport System*). O processo consiste em processar as imagens para obter curvas de luz, ou seja, a medida do fluxo de um objeto em relação ao tempo. As curvas serão utilizadas para a detecção das estrelas variáveis e a determinação de parâmetros físicos das mesmas.

3 Material e Métodos

O projeto teve três principais frentes: as campanhas de observação; o desenvolvimento do *software* proposto; e a exploração de resultados alternativos que poderiam ser obtidos com os dados coletados. A proposta do trabalho veio a se consolidar na etapa de observações.

3.1 Observações

Durante o desenvolvimento do projetos foram realizadas mais campanhas de observação além das que já haviam sido feitas, três no telescópio B&C e uma como projeto backup no telescópio PE.

No presente trabalho foram usados apenas dados obtidos com o telescópio B&C. A configuração instrumental usada no PE levou a uma complexidade maior na redução dos

dados. Portanto não foram usados até então.

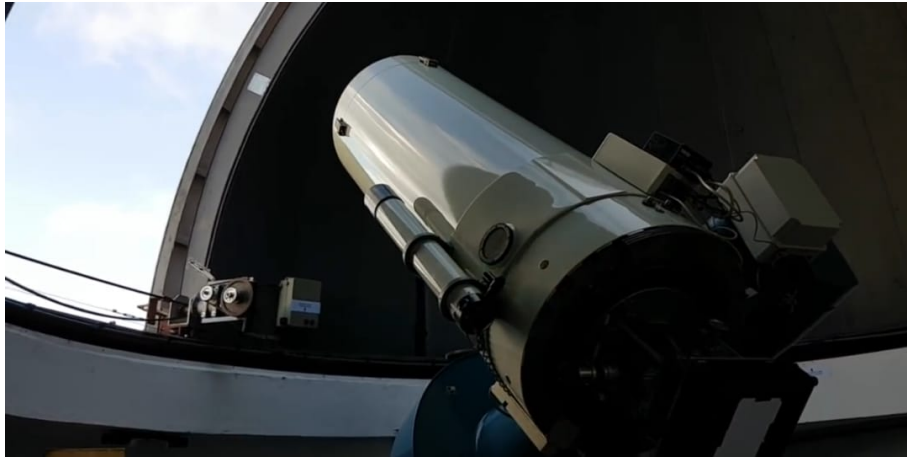


Figura 1: Telescópio B&C

O B&C foi equipado com um detector CCD (*Charge Coupled Device*) da Andor, modelo Ikon-L de 2048x2048 pixels, montados na Câmara Direta 2 (suporte para instrumentação), com uma roda de filtros UBVRI, e sistema de guiagem. Esta configuração instrumental resultou numa escala de placa de 0.34" por pixel, e num campo de visão de 11.6'x11.6'.

Seguindo a configuração adotadas para o CCD nas missões anteriores foi usada uma taxa de leitura de 1 MHz e pré amplificação de 1x, em modo de alta sensibilidade, resultando num tempo de leitura de 7 segundos um nível de contagem de base de 870 ADU e um ruído de contagem de base de 9 elétrons rms.

As observações foram realizadas tanto em modalidade remota quanto presencial. Em ambas situações no início da noite é feito o seguinte procedimento:

1. Checar condições climáticas (Boletim e sensores);
2. Ligar sistema de refrigeração do CCD;
3. Atualizar metadados (cabecalho e estrutura de pastas);
4. Criar documentos de log (reunir informações sobre as condições de observação)
5. Aguardar câmera atingir temperatura operacional;
6. Tirar imagens de calibração (Bias e Flats de cúpula);
7. Apontar em estrela brilhante para ajustar foco e alinhamento do telescópio;

8. Iniciar observação dos alvos.

Durante toda a noite as imagens são inspecionadas através do computador de análise disponível no observatório. Para isso foi usado o IRAF em conjunto com o visualizador ds9 através dos comandos da tarefa imexamine. Enquanto isso, informações sobre a noite são registradas no log criado no início da noite.

Mais detalhes sobre o processo de observação estão disponíveis no relatório referente ao primeiro ano do projeto. Disponível no link:

WildDuckPipe: Relatório 2018-2019

3.2 Desenvolvimento do software

O *software* foi desenvolvido totalmente na linguagem de programação Python. Foram usados os pacotes disponíveis no ecossistema científico da linguagem, como o Numpy, Scipy, Pandas, Matplotlib e especialmente os pacotes afiliados do projeto Astropy, o qual contém as funcionalidades para manipulação de imagens FITS e para a fotometria. Sendo assim, foi desnecessário o uso de pacotes externos.

Durante o processo foram usadas as convenções da comunidade do Python. Foi decidido isso para garantir a interoperabilidade com outros pacotes e para facilitar futuras colaborações. Essas convenções estão descritas nas PEP's (Python Enhancement Proposals), em especial a PEP 8. Todas as convenções adotadas, tanto para programação quanto para o formato dos dados foram sintetizadas no documento disponível no link abaixo.

WildDuckPipe: Design Doc

O processo de exploração das idéias e desenvolvimento das rotinas foram feitas dentro de cadernos do Jupyter. Um caderno Jupyter é uma plataforma que permite escrever códigos em Python dentro de uma estrutura de células. É possível escrever células de texto e de código, o que facilitou a documentação do processo. Os cadernos estão disponíveis no link abaixo.

WildDuckPipe: Jupyter Notebooks

O código final foi desenvolvido usando Git e hospedado no GitHub. O Git é um sistema de controle de versionamento, usado para registrar as mudanças sendo feitas no código, e combinar versões diferentes. A ferramenta ajuda não só com a documentação do processo, como também facilita colaboração e o conserto de *bugs*. Já o GitHub é uma plataforma *online* para o uso do Git, que permite a hospedagem do projeto. Ele está disponível no link abaixo.

<https://github.com/jpssrocha/WildDuckPipe>

3.3 Detalhes de implementação (Rotina principal)

Durante o processo de implementação foram usados diagramas para ter uma visão geral do processo. O programa começa com a calibração das imagens ou pré-processamento.

3.3.1 Calibração das imagens

Esta primeira etapa consiste em organizar as imagens brutas, gerar imagens de calibração e aplicá-las nas imagens de ciência. Nessa etapa o programa parte da pasta inicial com os dados brutos e entrega os arquivos de calibração e as imagens de ciência calibradas, organizados numa estrutura de pastas.

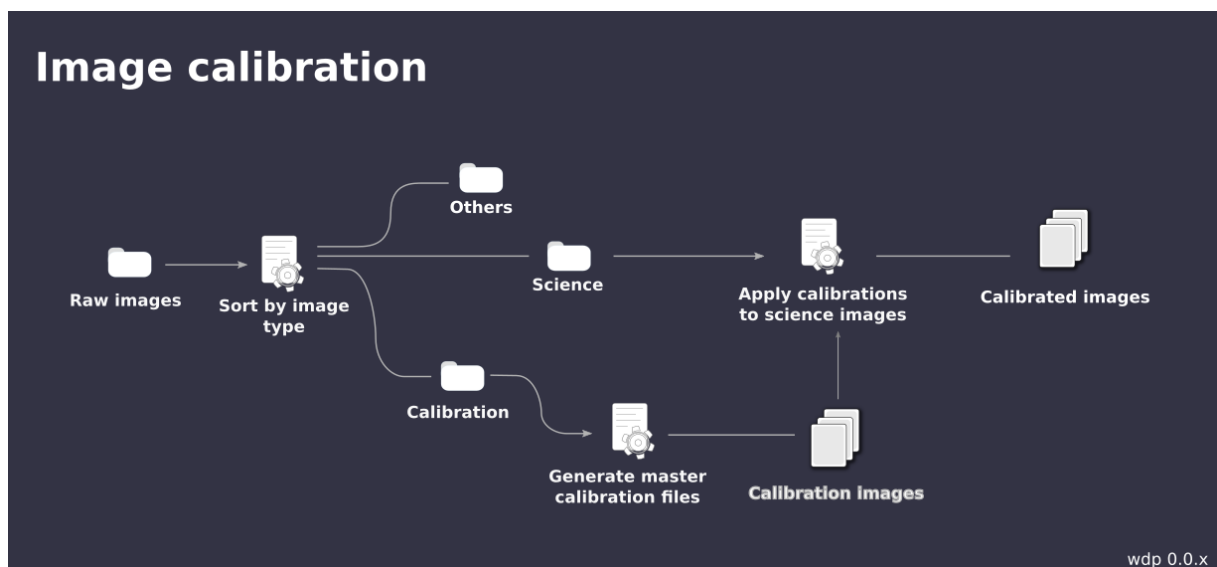


Figura 2: Fluxograma do algoritmo de calibração inicial

As imagens são inicialmente organizadas por tipo. Para isso é usado o módulo de I/O do Astropy. São lidos os cabeçalhos das imagens FITS. A partir das palavras chave do

cabeçalho é criada uma estrutura de pastas usando o módulo OS da biblioteca padrão do Python. Com isso o algoritmo usa o módulo OS para mover os arquivos para as devidas pastas.

Com isso o programa segue criando as imagens de calibração mestras (*Master*) de *Bias* e *Flat*. Primeiro as matrizes das imagens são carregadas, e em seguida combinadas pela mediana usando as operações vetorizadas do pacote Numpy, o que resulta no *Master Bias*. O *Master Bias* é então subtraído das imagens de *Flat* que, por sua vez são combinadas pela mediana da mesma forma que o *Bias*. É tirada a média da estatística dos pixels da imagem resultante, e os pixels são normalizados usando esse valor. Dessa forma são feitos os *Master Flats*, que são gravados no disco.

O próximo passo consiste em aplicar as calibrações nas imagens de ciência. Aplicar as calibrações numa imagem é basicamente resolver a seguinte equação para cada pixel:

$$I(x, y) = S(x, y) \times F(x, y) + B(x, y) \quad (1)$$

Onde I é o valor dos pixels na imagem, S o sinal, B o *Bias* e F o *Flat*, x e y indicam a a posição do pixel na matriz, x a coluna e y a linha. O processo de calibração consiste em resolver essa equação para o sinal S . Para isso é subtraído o *Bias* pixel a pixel, depois dividido pelo *Flat*. Assim são obtidas imagens calibradas, levando-se em conta os efeitos instrumentais. Desta forma as informações contidas em cada imagem são proporcionais ao sinal físico dos objetos astronômicos.

3.3.2 Extração de parâmetros

Para achar imagens com defeitos, como nuvens, foco ruim e outros, são extraídas as estatísticas básicas e parâmetros que indicam a qualidade da imagem. Nesse processo indicamos as imagens que serão analisadas e com isso é montada uma tabela de parâmetros. É usada uma imagem de referência para fazer um alinhamento. Nessa imagem são apontadas coordenadas para calcular parâmetros do céu e das estrelas. A partir disso as imagens são alinhadas e os parâmetros são calculados para cada imagem. Essas informações são usadas para montar uma tabela.

A imagem de referência é usada para o alinhamento. Para isso é usado o pacote AstroAlign, afiliado do projeto Astropy. O pacote acha as transformações de um *frame* para o outro comparando triângulos formados a partir das fontes pontuais nas imagens.

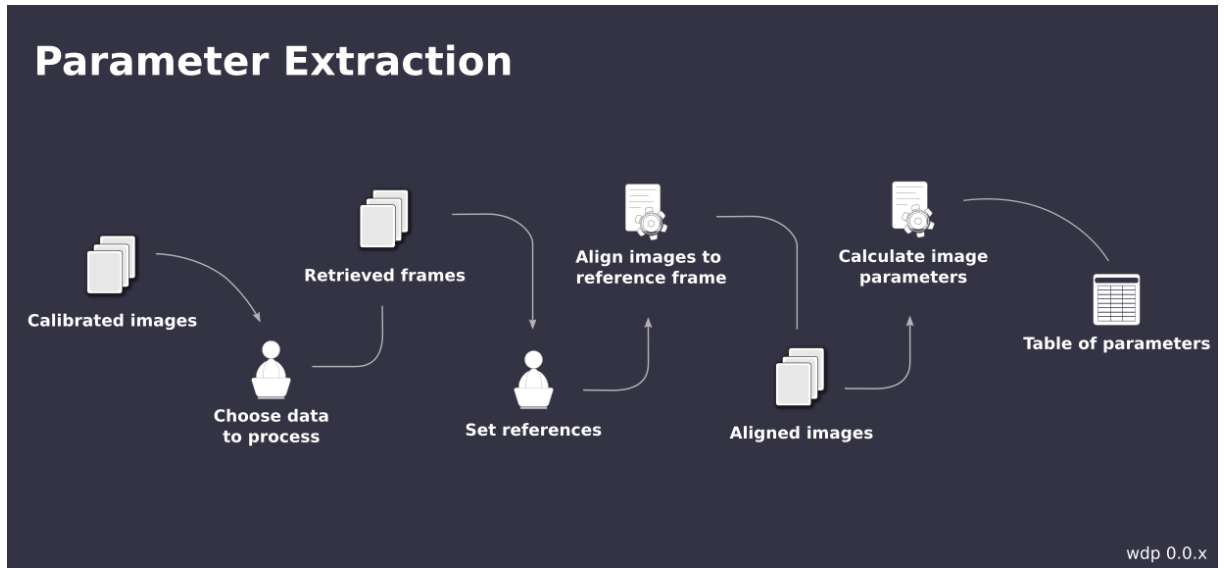


Figura 3: Fluxograma do algoritmo de extração de parâmetros

As imagens são comparadas uma a uma ao *frame* de referência. Ao fim do processo são salvas as imagens após a aplicação das transformações.

Das imagens alinhadas são extraídas as estatísticas dos pixels. São usadas as funções do Numpy para extrair das matrizes as seguintes estatísticas: máximo, mínimo, média, mediana e desvio padrão. Com as informações é iniciado um *dataframe* do pacote Pandas, para guardar as informações.

Usando das coordenadas indicadas para o céu são extraídas matrizes (basicamente caixas ao redor das posições indicadas) de locais sem estrelas, e delas são calculados a mediana e o desvio padrão do fundo do céu. Isso é feito usando as funções do Numpy. Com os valores calculados para cada pedaço é tirada uma média, que serão usadas como estimativas final.

Na sequência, usando as coordenadas das estrelas, são extraídas caixas com estrelas para calcular a *Full Width at Half Maxima* (FWHM). O processo é semelhante ao feito pelo IRAF na tarefa *imexamine*. Primeiro é calculado o centroide da distribuição, para isso são usadas as distribuições marginais em x e y, é ajustada uma gaussiana em cada distribuição e a posição do pico em cada eixo é usada como centro. Depois a distância ao centroide é calculada para cada pixel. Por fim é ajustada uma função de Moffat (eq. 2) à distribuição do número de contagens em função da distância ao centroide de cada pixel. Da função ajustada é extraída a FWHM (eq. 3).

$$f_{Moffat}(r) = A \left[1 + \left(\frac{r - r_0}{R} \right)^2 \right]^{-\beta} \quad (2)$$

$$FWHM = 2R\sqrt{2^{1/\beta} - 1} \quad (3)$$

Esses parâmetros são calculados para cada imagem e por fim montados numa tabela, que é salva no formato CSV (*Comma Separated Values*). Essa tabela será usada para remover *outliers* e ajudar na inspeção dos dados.

3.3.3 Filtragem e combinação

Com os parâmetros calculados é possível filtrar as imagens que apresentam algum problema, e as imagens restantes podem ser combinadas.

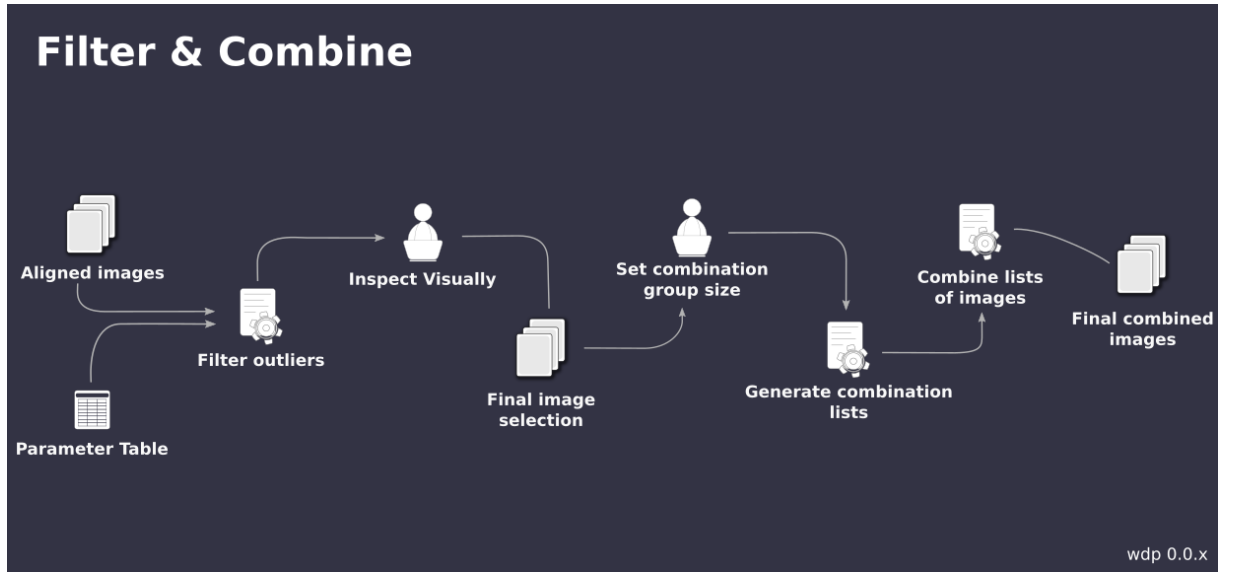


Figura 4: Fluxograma do algoritmo de seleção e combinação de imagens

Foi observado enquanto explorando os parâmetros que os histogramas seguem uma distribuição gaussiana, portanto é possível filtrar *outliers* usando o desvio padrão. Os parâmetros usados para a filtragem foram o brilho do fundo do céu, o ruído do céu e o FWHM. São separadas todas as imagens que não estejam dentro de 3 desvios padrão de distância da média. Com isso sobram menos imagens para inspecionar visualmente.

A inspeção visual é feita com a ajuda de gráficos dos parâmetros visualizando-os em função do tempo e com o visualizador ds9 para ver as imagens. O gráfico ajuda a identificar em quais pontos da noite os parâmetros variaram além do esperado. Depois as

imagens identificadas são visualizadas para procurar problemas. Caso seja achado algum problema a imagem é separada das demais. As imagens restantes são passadas para a tarefa de combinação.

Passando a quantidade de imagens a serem combinadas o algoritmo de combinação agrupa as imagens em listas e combina cada lista gerando as imagens finais. Primeiramente a tarefa verifica quais imagens estão próximas no tempo, imagens cujo intervalo de tempo entre elas seja maior que 5 vezes o tempo de exposição não serão colocadas na mesma lista. Com isso são montadas as listas com a quantidade de imagens escolhida. Cada lista é passada para uma função que combina as imagens pela mediana e grava em disco o resultado final. Com isso as imagens finais estão prontas para a fotometria.

3.3.4 Montagem das curvas de luz

Com as imagens finais é feita a fotometria imagem por imagem, que por fim é usada para montar uma tabela com as curvas de luz. Os parâmetros são re calculados para as imagens finais. Com isso e a imagem de referência é feita a detecção das estrelas. Com isso é montado um catálogo com as posições. Esse catálogo e a tabela de parâmetros são passados para uma função que irá fazer fotometria por abertura em cada imagem. Por fim são geradas várias tabelas de fotometria que são usadas para construir a tabela final das curvas de luz.

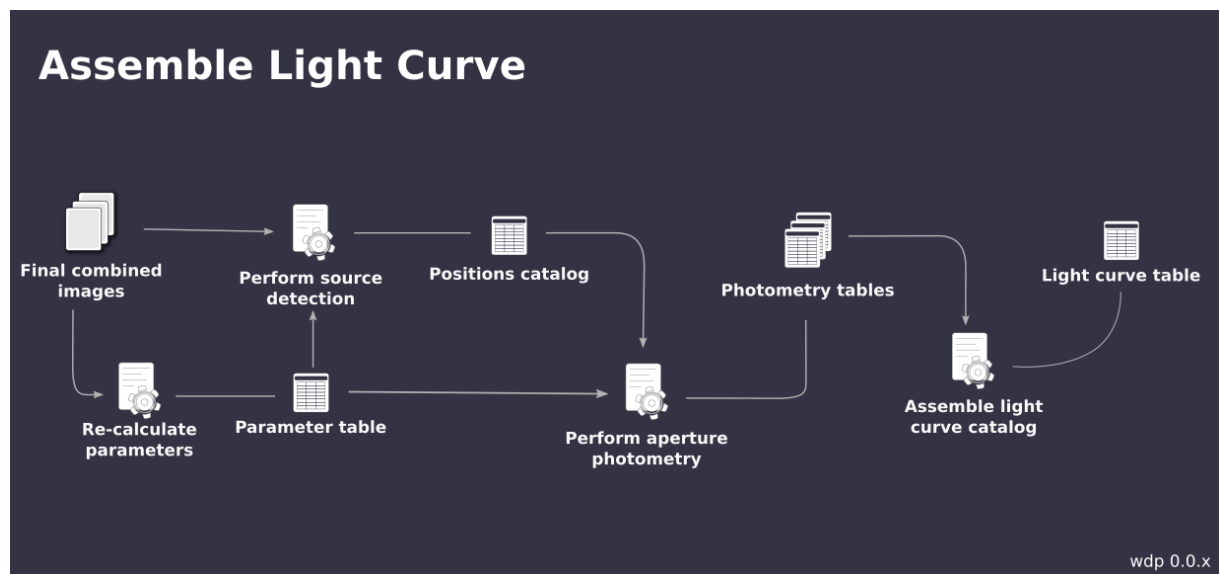


Figura 5: Fluxograma do algoritmo de montagem da tabela das curvas de luz

Os parâmetros das imagens finais são calculados pois serão utilizados na fotometria. O

procedimento é igual ao feito anteriormente, reaproveitando as definições passadas. Com isso uma segunda tabela de parâmetros é gerada.

É recuperado da tabela as informações da imagem de referência, que será usada para detectar as estrelas. Para isso é usada uma implementação do DAOFIND (Stetson 1987) disponível no pacote photutils, afiliado do projeto astropy. É passado para o algoritmo o valor do fundo do céu subtraído do ruído do fundo do céu que será usado como limite de detecção (*threshold*) e o FWHM. Como resultado é retornado um catálogo de posições para as fontes na imagem, que é escrito em disco como uma tabela CSV.

Com as posições, parâmetros e imagens finais é possível prosseguir com a fotometria por abertura, que será realizada em cada imagem para gerar tabelas de fotometria para cada uma. A fotometria por abertura foi feita usando uma implementação semelhante ao do IRAF. O algoritmo foi montado usando ferramentas do photutils adaptado de rotinas disponíveis no GitHub do STScI feitas para a WFC3 (*Wide Field Camera 3*) por Varun Bajaj. O algoritmo posiciona as aberturas para a estrela e os anéis do céu. A partir do anel do céu é estimado o ruído local que é subtraído das contagens dos pixels na abertura da estrela. Os pixels dentro da abertura são somados, e a soma é dividida pelo tempo de exposição e área da abertura, obtendo o fluxo na abertura. O valor final é convertido para magnitude usando a formula de Pogson, que pode ser vista na eq. 4, onde F é o fluxo em contagens (ADU) somado a um valor aproximado do ponto zero da fotometria (foi utilizado 25). O processo é feito para todas as estrelas no catálogo usando a FWHM para escalar o tamanho das aberturas em cada imagem. O processo gera tabelas individuais de fotometria para cada imagem.

$$M = 25 - 2.512 \log_{10}(F) \quad (4)$$

As tabelas de fotometria não são escritas em disco, primeiro é montada uma tabela final, a tabela com as curvas de luz, ou seja a magnitude calculada para cada imagem avançando no tempo. Das tabelas de fotometria são extraídas a posição em relação as colunas e linhas do CCD, bem como o número de identificação do catalogo de posições. Com isso são extraídas a magnitude e o erro estimado para cada cada imagem. Assim as informações são concatenadas numa tabela que contém a informação de como a magnitude das estrelas varia em relação ao tempo, as chamadas curvas de luz. Essa tabela é gravada no disco e usando as informações dela será feita a fotometria diferencial.

3.3.5 Fotometria diferencial

Para reduzir o ruído na curva de luz é usada a técnica de fotometria diferencial, que consiste em comparar o brilho das estrelas dentro de cada imagem entre si, gerando uma curva de luz que sofre menos com a variabilidade das condições de observação. Isso é feito a partir da geração de uma magnitude de referência para cada imagem. Essa referência é obtida das estrelas cuja variabilidade entre imagens é menor. Por fim é tirada a diferença entre a magnitude de referência e a de cada estrela. Como resultado é obtida uma curva de luz que tem menos influência da variação das condições da observação e da instrumentação.

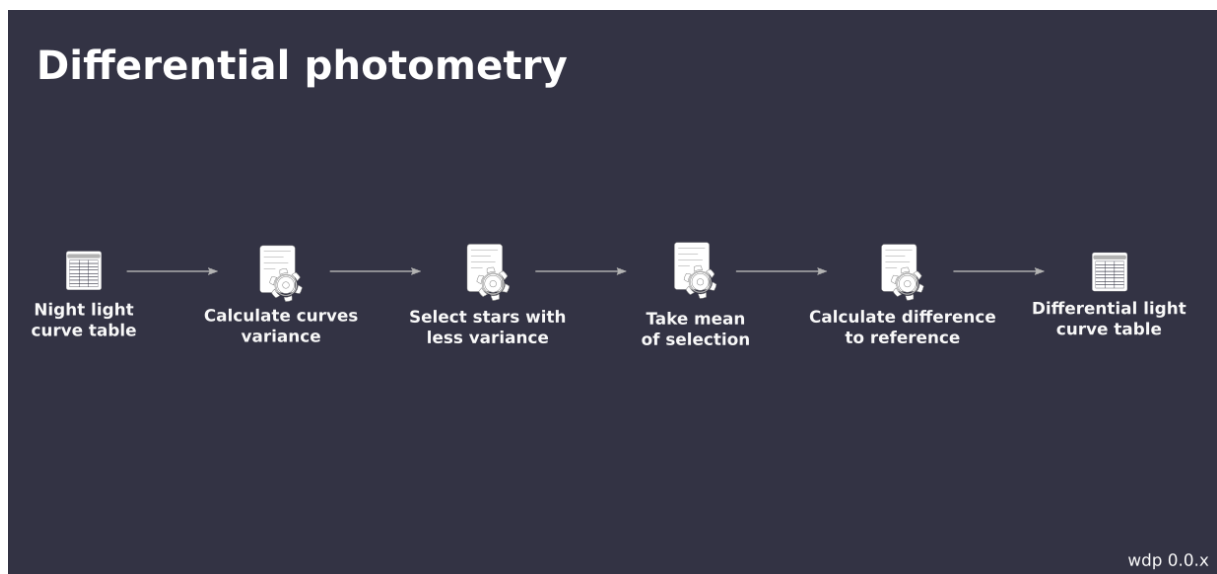


Figura 6: Fluxograma do algoritmo de fotometria diferencial

Para a seleção das estrelas que variam menos, é carregada a tabela com as curvas de luz, e calculado o desvio padrão para cada estrela em relação ao tempo. Para isso as curvas de luz são carregadas usando o pacote Numpy. O Numpy tem funções especiais para trabalhar com os valores faltantes, que aparecem quando o algoritmo de fotometria falha. Os dados são extraídos usando a notação de acesso vetorial do Numpy e posteriormente calculadas as operações em eixos. Com isso é gerado um vetor com os valores do desvio padrão para cada estrela durante a noite.

O vetor de valores de desvio é ordenado em ordem crescente e são selecionadas 100 estrelas que variam menos, das quais são recuperadas as curvas de luz. A reordenação é feita com a função `sort` do Numpy. Depois de extraídos os valores, é criada uma máscara usando a função `where`, apontando o índice original da estrela. Com a máscara são

recuperadas as curvas originais.

A partir das curvas de luz das estrelas de referência, as que variam menos, é tirada uma média do brilho das estrelas em cada imagem. Cada coluna representa as estrelas em uma imagem, as médias são calculadas aplicando a função de média no eixo das colunas. Ao colapsar o eixo das colunas pela média é obtida magnitude de referência para cada imagem.

As magnitudes de referência são subtraídas, imagem por imagem, gerando a curva de luz final que será gravada em disco. Para isso é usada a capacidade de "broadcasting" do Numpy, que permite que a operação seja expressa como uma simples subtração em uma seleção de dados. Após a subtração é obtida a tabela final de curvas de luz que é gravada.

Essa tabela é o produto final da redução.

3.4 Experiência usando IRAF

Uma tentativa de verificar a viabilidade de usar o IRAF para a realização de reduções rápidas foi feita, para verificar que tipo de resultados poderiam ser obtidos em um dia usando o IRAF.

Primeiramente foram gerados os arquivos de calibração usando as tarefas especializadas do IRAF, zerocombine e flatcombine, que foram aplicadas nas imagens usando a tarefa ccdproc. As imagens calibradas foram analisadas visualmente através do ds9, e as imagens problemáticas foram separadas.

Foi escolhida uma imagem de referência, e todas as imagens foram alinhadas com a referência usando a tarefa imalign. As imagens foram combinadas 3 a 3 pela mediana usando a tarefa imcombine. Depois foram coletados os parâmetros da imagem de referência e ela foi usada com a tarefa daofind para gerar um catálogo geral com a posição das estrelas.

A coleta de parâmetros para a fotometria foi simplificada por tomar muito tempo para ser feita manualmente. Portanto as imagens foram agrupadas em quatro grupos e os parâmetros foram calculados para algumas imagens em cada grupo, sendo usado no final uma média dos valores achados em cada grupo.

Por fim, foi feita a fotometria por abertura usando a tarefa phot e as informações coletadas para cada imagem seguindo o mesmo procedimento que já vinha sendo utilizado. A tabela final e a fotometria diferencial foram feitas usando os *scripts* em Python que já

havia sido feitos no projeto anterior.

3.5 Diagramas de média por desvio

Para avaliar os resultados das curvas de luz, foi visualizado o gráfico de desvio padrão da curva em função da magnitude. Com as curvas de luz foi calculada para todas as estrelas o desvio padrão e a magnitude média em relação ao tempo. Com os valores obtidos foram montados os diagramas.

Nesse caso a tendência natural a ser seguida é que o desvio aumente com o aumento da magnitude, ou seja, quanto menor o brilho maior a incerteza na medida. Isso ocorre, pois com a diminuição do brilho da fonte o ruído do céu se torna cada vez mais importante. Outros pequenos efeitos também passam a ser percebidos mais facilmente. Portanto a tendência é sempre que o erro aumente com a magnitude.

O espalhamento nesse gráfico também é sensível a efeitos de variabilidade das próprias fontes e das condições de observação, o que é refletido na forma com que é feito o processo de fotometria. Um gráfico muito espalhado pode indicar que muito ruído foi levado em conta no processo de fotometria. No caso de uma fonte intrinsecamente variável, a mesma terá a tendência de apresentar naturalmente um desvio maior. Também existem aumentos que podem aparecer pela própria densidade de estrelas na imagem. Esses efeitos acabam se refletindo nesse gráfico, que pode ser usado como guia ao refinar o processo de redução e portanto foram utilizados para obter uma noção geral do resultado final.

4 Resultados e Discussão

Além dos dados coletados foi possível obter resultados não esperados com a ajuda do software. O mesmo possibilitou a realização de testes rapidamente o que ajudou na melhoria da qualidade da técnica de redução.

4.1 Dados coletados

Durante o projeto foram realizadas mais 4 campanhas de observação. Foram coletados dados para a validação dos métodos e para a procura de variáveis desconhecidas.

Foram consideradas noites usáveis as em que foram obtidas imagens sinalizadas como imagem de ciência.

Tabela 1: Volume de dados por missão

Missão	Noites usáveis/concedidas	Volume de Dados (Gb)	Alvo Principal
OI2019A-031	10/18	37.0	M41/NGC6067/M11
OI2019B-019	1/3	1.4	NGC6067
OP2019B-008	1/4	3.7	M11
OI2020A-033	6/16	70.0	NGC4755

Para a validação dos métodos foi observado M11 por já ter sido extensivamente estudado quanto ao aspecto de variabilidade, sendo inclusive um campo de calibração (por Peter B. Stetson). Foram obtidos dados suficientes para os testes.

Como é possível ver na tabela 1 foram obtidos mais de 100 Gb de dados brutos. Isso foi o que chamou a atenção para a necessidade do desenvolvimento de rotinas automatizadas para a redução de dados.

4.2 Software de redução

Os testes usando o *software* de redução ajudaram a alcançar resultados que não eram esperados. Por aumentar muito a agilidade foi possível otimizar os parâmetros, pois foi possível realizar vários testes para aprender sobre os aspectos do processo de geração das curvas de luz.

A primeira confirmação de que as tarefas de fotometria estavam funcionando corretamente foi com a construção de um diagrama cor magnitude.

4.2.1 Diagrama cor-magnitude

Durante o desenvolvimento das rotinas de fotometria o teste usado foi a criação de um diagrama cor-magnitude (CMD), por já conhecer o formato o mesmo deveria apresentar.

No diagrama é possível identificar facilmente a sequencia principal à esquerda, bem como ver as gigantes vermelhas na parte superior à direita.

Apesar da fotometria não estar calibrada, reproduz muito bem a morfologia do diagrama calibrado encontrado num trabalho de Terndrup & Pinsonneault (2007, ApJ 671, 1640). O que serviu de indicativo de que as rotinas de tratamento de imagem e fotometria estão funcionando de maneira adequada.

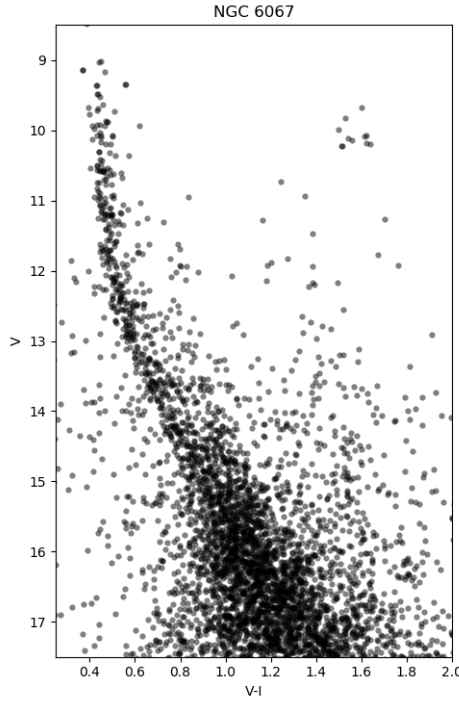


Figura 7: Diagrama cor magnitude de NGC6067

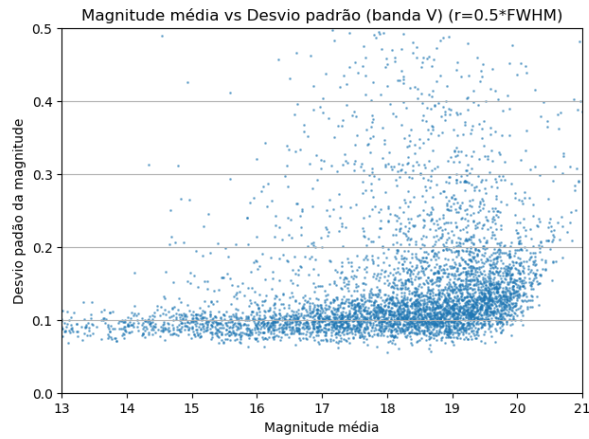
4.2.2 Otimização de abertura

Com o desenvolvimento do *software* surgiu a possibilidade de usar uma metodologia mais pragmática para a escolha do tamanho das aberturas para a fotometria. Em geral isso é escolhido de maneira subjetiva de acordo com a densidade do campo. No entanto com a automação da rotina de fotometria foi possível rodar testes com várias configurações diferentes e escolher a configuração que entregasse o melhor resultado.

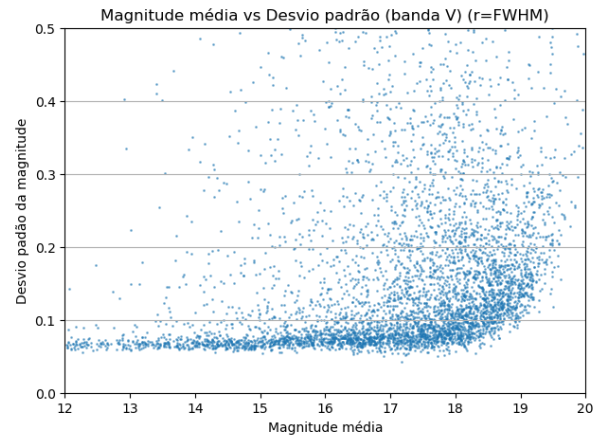
Na figura 8 temos vários gráficos, onde é possível verificar a variância nas curvas de luz, foi possível refazer as curvas de com várias combinações diferentes de parâmetros para a fotometria por abertura.

Foi observado um aumento na base do espalhamento com a diminuição do tamanho da abertura, no entanto uma diminuição do espalhamento. Já com o aumento da abertura foi observado o contrário, diminuição na base porém aumento do espalhamento em geral. Foi observado que isso ocorre devido ao aumento do ruído causado pelo céu, que é aumentado pelo campo de M11 ser bastante rico.

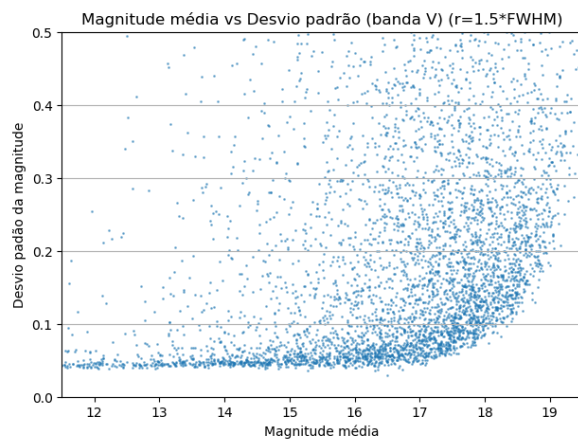
Por isso foi escolhida uma configuração intermediária, de 1.5 vezes a FWHM da imagem. Isso diminui a influência o céu enquanto mantém a base do espalhamento menor. No futuro isso deve ser otimizado, por hora é necessária uma base abaixo de 0.1 magni-



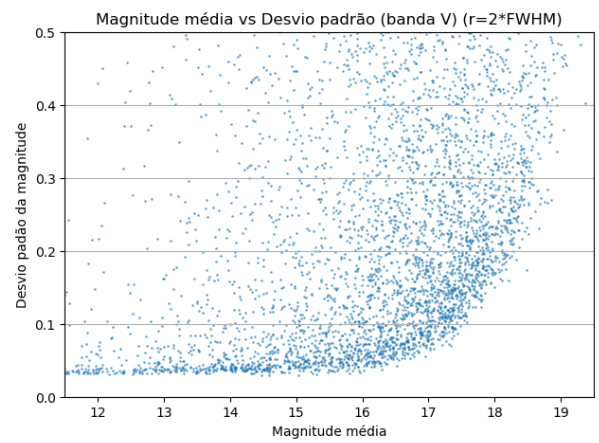
(a) $0.5 * \text{FWHM}$



(b) $1.0 * \text{FWHM}$



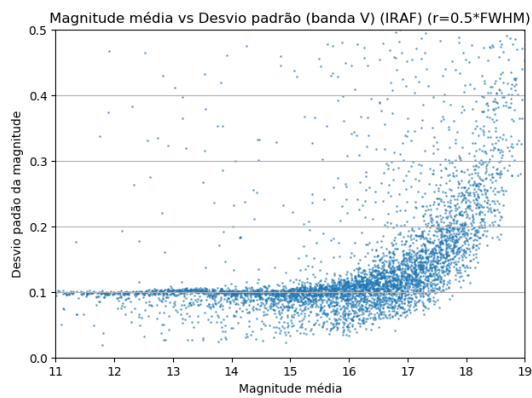
(c) $1.5 * \text{FWHM}$



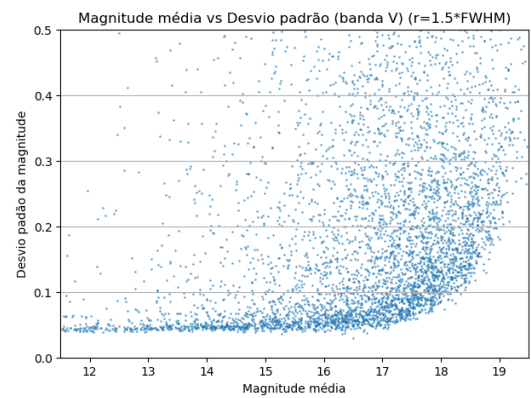
(d) $2.0 * \text{FWHM}$

Figura 8: Dispersão em diferentes aberturas

tude pois com a fotometria diferencial é possível atingir uma base baixa o suficiente para observar os fenômenos de variabilidade que estamos interessados.



(a) Redução manual



(b) Redução assistida

Figura 9: Espalhamento final

É possível observar que na fotometria sem o controle dos parâmetros aparece uma

distribuição artificial. A tendência que deveria ser observada, teoricamente, deveria ser uma curva que cresce com o aumento da magnitude. No entanto aparecem pontos cuja o erro diminui com o aumento. Isso indica que os parâmetros utilizados foram mal escolhidos ou houveram erros ao rodar a fotometria, no entanto não foi percebido sem a possibilidade dos testes rápidos. A distribuição com a redução assistida, reflete melhor o esperado.

Com isso é possível ver que a possibilidade da realização dos testes rápidos ajuda a melhorar bastante o resultado final. Na figura 9 podemos ver que a base do espalhamento da redução manual foi aproximadamente o dobro do obtido na redução assistida (0.1 para 0.05 magnitude).

Com o resultado otimizado em mãos foi possível obter uma base de erro ainda menor com a fotometria diferencial.

4.2.3 Fotometria diferencial

Com o maior controle sobre os parâmetros da redução dos dados e a possibilidade de realizar vários testes foi possível facilmente otimizar os parâmetros para obter um resultado menos ruidoso. Ao passar pelo algoritmo de fotometria diferencial um resultado melhor que os passados foi alcançado.

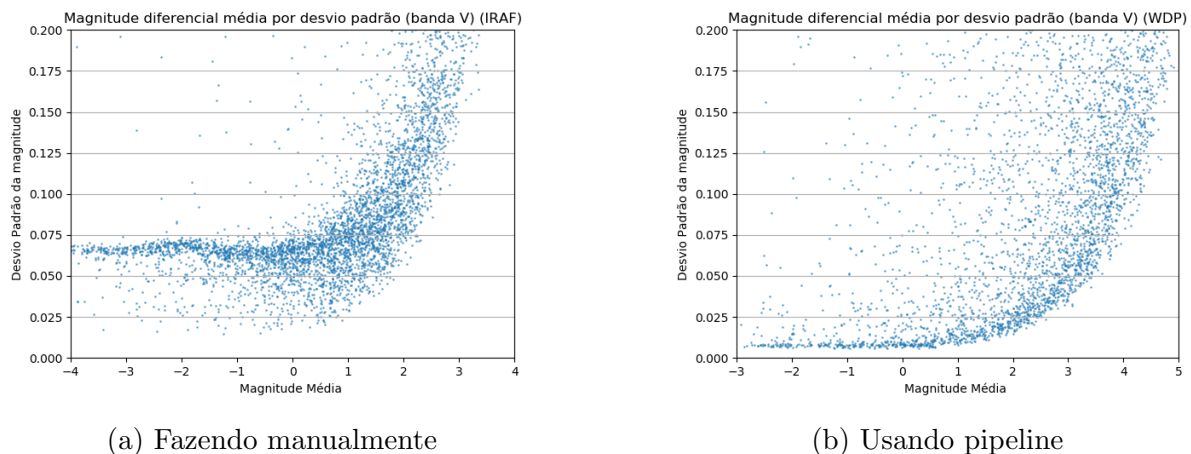


Figura 10: Espalhamento na fotometria diferencial

A base do espalhamento ficou em aproximadamente 0.06 magnitude no caso da redução com o IRAF e por volta de 0.007 no caso do uso do *pipeline*. Quase 9 vezes menor. Comparado com o melhor resultado feito manualmente usando IRAF, que foi apresentado no relatório passado de 0.04 magnitude tivemos um resultado quase 6 vezes melhor.

5 Conclusões

Durante o projeto foi necessário fazer um desvio grande para solucionar o problema gerado pela crescente quantidade de dados, com isso foi construída uma série de rotinas, que estão sendo integradas em uma *pipeline*, isso aumentou exponencialmente a agilidade na redução dos dados, fazendo com que um processo que levaria dias para fazer de forma manual, possa ser realizado em poucas horas.

Com isso, a maior velocidade para fazer ajustes e testes fizeram com que a qualidade do resultado final se tornasse muito maior do que o melhor resultado obtido manualmente (0.04 mag de base de erro para 0.006 mag) o que torna possível a observação de fenômeno de variabilidade de amplitudes muito menores.

Dessa forma podemos avançar com a criação de um catálogo geral usando todo os dados obtidos até então para detectar as variáveis. E durante o processo será possível refinar os processo de redução pela introdução de técnicas mais sofisticadas, como por exemplo, a fotometria por PSF.

6 Referências

- Sánchez, E. & DES Collaboration , 2010, JPhCS, 259, 2080
Sánchez, J. et al. 2020, MNRAS, 497, 210
Astropy Collaboration et al. 2013, A&A 558, 33
Stetson, 1987, PASP, 99, 191
Terndrup & Pinsonneault, 2007, ApJ, 671, 1640
Barret, 2001, ASPC, 216, 685B
Kerber et al. 2018, ApJ, 853, 15
Kerber et al. 2019, MNRAS, 484, 5530
Salinas, R. 2018, MmSai, 89, 101
Roelens et al. 2016, CSSS Conference.

7 Publicações

Não houveram publicações