



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

PROTOCOLO DE AVALIAÇÃO DIAGNÓSTICA DA ASSIMILAÇÃO DE DADOS DE RADIÂNCIA NO SMNA

CAROLINE VIEZEL
LUIZ F. SAPUCCI

Relatório técnico da primeira avaliação de desempenho da assimilação de dados de radiância no Sistema de Modelagem Numérica e Assimilação (SMNA) do CPTEC usando o ReadDiag, no contexto do subprojeto 9.1 do Programa de Capacitação Institucional (PCI-INPE).

URL do documento original:

[<http://urlib.net/>](http://urlib.net/)

INPE
Cachoeira Paulista
Dezembro de 2024

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3945-6923/6921

Fax: (012) 3945-6919

E-mail: <pubtc@sid.inpe.br>

**COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO
DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):****Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Amauri Silva Montes - Coordenação Engenharia e Tecnologia Espaciais (ETE)

Dr. André de Castro Milone - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Joaquim José Barroso de Castro - Centro de Tecnologias Espaciais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Drª Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

PROTOCOLO DE AVALIAÇÃO DIAGNÓSTICA DA ASSIMILAÇÃO DE DADOS DE RADIÂNCIA NO SMNA

CAROLINE VIEZEL
LUIZ F. SAPUCCI

Relatório técnico da primeira avaliação de desempenho da assimilação de dados de radiância no Sistema de Modelagem Numérica e Assimilação (SMNA) do CPTEC usando o ReadDiag, no contexto do subprojeto 9.1 do Programa de Capacitação Institucional (PCI-INPE).

URL do documento original:

[<http://urlib.net/>](http://urlib.net/)

INPE
Cachoeira Paulista
Dezembro de 2024

Resumo

O Sistema de Modelagem Numérica e Assimilação de dados (SMNA), que está em modo pre operacional no INPE, composto pelo modelo de previsão do tempo BAM (*Brazilian Global Atmospheric Model*) e o sistema de assimilação *Gridpoint Statistical Interpolation* (GSI), o qual permite explorar os dados provenientes de redes de observação convencionais e de satélites, visando gerar previsões de tempo mais precisas. A assimilação dos dados de satélite, como a radiância, compõem uma base de dados de grande densidade para a representação das características da superfície no modelo de previsão. Entender como a assimilação dos dados de radiância está impactando as previsões fornecidas pelo SMNA possibilita indicar melhorias e fazer adequações afim de proporcionar aproveitamento máximo dessas observações fornecidas pelos satélites. No SMNA os arquivos de diagnóstico do processo de assimilação fornecidos pelo GSI são lidos e analisados pela plataforma **readDiag**. O presente trabalho descreve uma versão inicial do protocolo de avaliação da assimilação de radiâncias no readDiag, que está disponível para os diferentes usuários da assimilação de dados. Visando exemplificar o protocolo como uma prova de conceito, esse documento mostra resultados de um estudo realizado com experimentos da versão 2.4.0 em desenvolvimento, com um experimento gerado através da versão operacional 2.3.1. Trata-se da primeira avaliação da assimilação de dados de radiâncias do SMNA, cujos resultados visam contribuir com os esforços de aprimoramento das futuras versões do sistema.

Palavras chave: SMNA; Radiância; GSI; Previsão Numérica de Tempo, Assimilação de Dados.

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Tabela com as flags do <code>idqc</code> de 0 à 12 e as flags específicas dos sensores AMSU-A e HIRS/4.	9
2.2 Esquema de classificação das observações de radiância com <code>iuse</code> e <code>idqc</code>	9
3.1 Exemplo de <i>diagrama de hovmoller</i> obtido através da função <code>time_series_radi</code> do readDiag	13
3.2 Exemplo de mapa gerado pela função <code>statcount</code> do readDiag , identificando os dados do canal 06 em assimilados (verde) e rejeitados (vermelho).	14
3.3 Exemplo de mapa gerado pela função <code>statcount</code> do readDiag , identificando os dados do canal 08 em monitorados-assimilados (teal) e monitorados-rejeitados (purple).	15
3.4 Exemplo da série temporal do total de dados assimilados, monitorados e rejeitados, gerado pela função <code>statcount</code> do readDiag para dados de radiância.	15
3.5 Exemplo de mapa gerado pela função <code>plot</code> do readDiag , identificando os dados do canal 13 em relação ao parâmetro OmA e fixando os valores de mínimo e máximo do intervalo de cores.	16
4.1 Série temporal das médias globais (1ª linha), desvio padrão (2ª linha) e quantidade de dados assimilados (3ª linha) dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-15 no período 2024020100 - 2024022918.	20
4.2 Distribuição espacial das observações assimiladas do sensor AMSU-A no satélite NOAA-15, canais: 1, 2, 3, 4, 5 e 15, em 2024021912.	21
4.3 Distribuição espacial das observações assimiladas e rejeitadas do sensor AMSU-A no satélite NOAA-15, canal 5, em 2024021912.	22
4.4 Série temporal das médias globais (1ª linha), desvio padrão (2ª linha) e quantidade de dados assimilados (3ª linha) dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-18 no período 2024020100 - 2024022918.	23
4.5 Série temporal das médias globais (1ª linha), desvio padrão (2ª linha) e quantidade de dados assimilados (3ª linha) dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-19 no período 2024020100 - 2024022918.	24

4.6	Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-15 nos experimentos 19 (a) e 20 (b), período 2024020100 - 2024022918.	25
4.7	Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-18 nos experimentos 19 (a) e 20 (b), período 2024020100 - 2024022918.	26
4.8	Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-19 nos experimentos 19 (a) e 20 (b), período 2024020100 - 2024022918.	27
4.9	Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-15 nos experimentos 19 (a) e 21 (b), período 2024030100 - 2024033118.	29
4.10	Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-18 nos experimentos 19 (a) e 21 (b), período 2024030100 - 2024033118.	30
4.11	Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-19 nos experimentos 19 (a) e 21 (b), período 2024030100 - 2024033118.	31
4.12	Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-15, canal 13, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.	33
4.13	Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-19, canal 13, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.	34

4.14	Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) obtidos pelo satélite NOAA-15, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b).	35
4.15	Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) obtidos pelo satélite NOAA-18, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b).	36
4.16	Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) obtidos pelo satélite NOAA-19, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b).	37
4.17	Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-18, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.	38
4.18	Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-19, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.	39
4.19	Série temporal das médias globais e dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelos satélite NOAA-18 (a) e NOAA-19 (b) do canal 14 no experimento 21.	40
4.20	Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) (esquerda) e série temporal das médias e desvio de OmF e OmA (direita) obtidos pelo satélite MetOp-b, canal 14, no experimento 20 (mês de Fevereiro) (a) e no experimento 21 (mês de Março) (b). . . .	41
4.21	Série temporal das médias globais e dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelos satélite MetOp-b, canal 14, no EXP20 (esquerda) e EXP21 (direita).	42

4.22 Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite MetOp-b nos experimentos 20 (a) e 21 (b). . 43

LISTA DE TABELAS

	<u>Pág.</u>
2.1 Descrição das flags <code>iuse</code>	8
4.1 Lista de experimentos realizados para avaliar a versão SMNA V2.4.0. . .	18
4.2 Classificação dos canais em todos os experimentos: assimilado, monito- rado e rejeitado.	32

SUMÁRIO

	<u>Pág.</u>
1 Introdução	2
1.1 Objetivos	2
1.2 Organização do relatório	3
2 Materiais e métodos	4
2.1 readDiag: descrição e estrutura	4
2.2 Aspectos da avaliação da assimilação de dados de Radiância	6
3 Assimilação de dados de Radiância: implementação	11
3.1 Função time_series_radi	11
3.2 Função statcount	13
3.3 Função plot	16
4 Resultados: análise dos dados de radiância	18
4.1 Comparação entre os EXP19 e EXP20	18
4.2 Comparação entre os EXP19 e EXP21	28
4.3 Análise por canal	32
4.3.1 Canal 13	33
4.3.2 Canal 14	35
4.4 Comparação entre os EXP20 e EXP21 do satélite MetOp-b	40
5 Considerações finais	44
REFERÊNCIAS BIBLIOGRÁFICAS	46
Notebook tutorial: readDiag radiância	47

1 Introdução

Adotado pelo Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) do Instituto Nacional de Pesquisas Espaciais (INPE), o sistema Gridpoint Statistical Interpolation, GSI, é responsável pela assimilação de dados fornecidos por estações meteorológicas de superfície instaladas no continente, de sensores em boias oceânicas e embarcados em navios, dados de ar superior provenientes de radiossondas, drop sondas, dados de sensores meteorológicos a bordo de aviões, dados de radiância estimados de sensores de satélites nas faixas de frequência infravermelha e micro-ondas, dados de rádio ocultação GNSS e dados de vento por satélites. A assimilação ocorre de forma direta no GSI que é integrado ao Brazilian Atmospheric Model (BAM). Ambos compõe o Sistema de Modelagem Numérica e Assimilação (SMNA) em pré operação no centro, responsável por fornecer condições iniciais para os modelos rodados no CPTEC, tornando-os independentes de outros centros.

Ao executar o GSI, arquivos de diagnóstico são gerados pelo sistema para avaliação do processo de assimilação de dados. Tais arquivos necessitam de um pós-processamento para análise do desempenho da assimilação. O pacote **readDiag** foi desenvolvido com o objetivo de facilitar o acesso aos arquivos diagnóstico e criar estruturas de dados adequadas para a sua manipulação e consequente análise. A versão do ReadDiag disponível para essa pesquisa estava apta para análise de todas as bases de dados, no entanto cada tipo de dados requer alguns ajustes e aprimoramento das funções disponíveis para adequar as análises para enaltecer alguns aspectos de cada tipo de dados. Para o caso da radiância, diferentes dos demais tipos de dados, um mesmo sensor tem diferentes tipos de dados em cada um de seus canais espectrais, o que exige uma abordagem apropriada para a avaliação desse tipo de dados.

1.1 Objetivos

O presente documento tem por objetivo apresentar a metodologia utilizada para adequar o readDiag para a avaliação dos dados de radiância, apresentar detalhes do protocolo de avaliação proposta para essa base de dados e resultados de dois experimentos do SMNA, para exemplificar o uso desse protocolo na avaliação das versões orientando as implementações na evolução do sistema. Sendo assim, o presente trabalho descreve modificações que foram realizadas na versão em desenvolvimento do **readDiag** V1.2.3 (disponível em <<https://github.com/GAD-DIMNT-CPTEC/readDiag/releases/tag/v1.2.3>>) as quais foram integradas na nova versão do readDiag (versão V1.3.2, disponível em <<https://github.com/GAD-DIMNT-CPTEC/readDiag/releases/tag/v1.3.2>>) sendo essa apta à leitura e análise do desempenho

do processo de assimilação dos dados de radiância, como fruto dessa pesquisa.

1.2 Organização do relatório

Na sequência deste capítulo introdutório, apresenta-se no capítulo 2 uma descrição d estruturação da ferramenta **readDiag** base para essa pesquisa e é apresentado informações relevantes para compreensão da organização dos arquivos de diagnóstico dos dados de radiância e metodologia utilizada para desenvolver funcionalidades específicas para esse tipo de dado dentro do **readDiag**. Neste capítulo, também descreve-se as variáveis de maior interesse para análise do processo de assimilação da radiância. O capítulo 3 descreve as modificações e implementações desenvolvidas na plataforma **readDiag** para avaliação do diagnóstico da radiância. Neste capítulo funções que foram modificadas são citadas e as funções criadas especificamente para análise dos dados de radiância são apresentadas. Na sequência, o capítulo 4 trás resultados do processo de análise dos dados de radiância gerados em três experimentos. Para finalizar o capítulo 5 apresenta as conclusões e considerações finais deste relatório, indicando futuros aprimoramentos que podem ser desenvolvidos para auxiliar no processo de avaliação da assimilação de dados de radiância. No Apêndice é apresentado um esquema passo-a-passo (notebook do python) de como utilizar as novas funções do **readDiag** desenvolvidas para a avaliação da assimilação de dados de radiância, que pode ser utilizado como material de treinamento para novos usuários.

2 Materiais e métodos

Para o desenvolvimento dessa pesquisa foi utilizada uma versão do `readDiag` na qual foi desenvolvido novas funcionalidades específicas para os dados de radiação que são exploradas no protocolo de avaliação aqui proposto. Nesse capítulo é apresentado uma descrição detalhada do `readDiag` e os aspectos da avaliação da assimilação dos dados de radiação e a metodologia utilizada nessa pesquisa.

2.1 `readDiag`: descrição e estrutura

Plataforma desenvolvida para leitura e análise de arquivos de diagnóstico do sistema GSI, o **`readDiag`** é uma interface Fortran/Python composta de duas classes:

- 1^a) classe `read_diag`: responsável pela leitura dos arquivos de diagnóstico.
- 2^a) classe `plot_diag`: responsável por fornecer ao usuário diferentes tipos de figuras para posterior análise do processo de assimilação.

O sistema GSI gera dois arquivos de diagnóstico que são lidos pelo **`readDiag`**. O primeiro arquivo contém a medida *Observation Minus Forecast* (OmF) relacionada ao primeiro outer loop (outer loop são as iterações externas do processo de minimização e linearização do GSI (OLIVEIRA et al., 2022)). O segundo arquivo contém a medida *Observation Minus Analysis* (OmA) relacionada ao último outer loop do processo de minimização desempenhado pelo GSI.

A classe `read_diag` contém scripts das funções que são responsáveis pela leitura dos arquivos de diagnóstico e fornece ferramentas para extração de informações pertinentes do conjunto de dados, como por exemplo, variável e tipo. Abaixo apresentam-se as funções que compõe esta classe e uma breve descrição do seu objetivo.

► `__init__`: realiza o processo de leitura dos arquivos de diagnóstico. Para cada tipo de dado, seja convencional ou de radiação, esta função realiza a leitura dos arquivos referentes ao primeiro (OmF) e último (OmA) outer loops e cria uma lista nomeada `gdf_list` com os dados dos dois arquivos de diagnóstico. Vale mencionar que para cada dia e horário do ciclo de assimilação os dois arquivos de diagnóstico são gerados pelo GSI e o **`readDiag`** executa de forma dinâmica a leitura dos arquivos no período designado previamente pelo usuário, agregando a lista `gdf_list` sublistas referentes a cada data e horário especificado. Como exemplo, considere que o usuário queira analisar os diagnósticos do processo de assimilação gerados

entre os dias 01/02/2024 e 02/02/2024. O GSI realiza o processo de assimilação nos horários sinóticos 00, 06, 12 e 18 UTC. Logo, a lista de dados `gdf_list` terá oito sublistas, sendo `gdf_list[0]` a sublista que armazena as informações referente ao dia 01/02/2024 às 00 UTC, `gdf_list[1]` a sublista referente ao dia 01/02/2024 às 06 UTC, e consequentemente a sublista `gdf_list[7]` contém os dados do dia 02/02/2024 às 18 UTC.

- **overview**: cria e retorna um dicionário com as informações existentes no arquivo.
- **pfileinfo**: realiza a impressão de uma lista das observações e tipos existentes no arquivo.
- **close**: fecha arquivos previamente abertos.
- **tocsv**: para a variável e tipos escolhidos pelo usuário, esta função constrói um arquivo CSV com valores de média e desvio padrão de OmF e OmA, além de fornecer a quantidade total de observações e data.

A classe *plot_diag* é constituída de funções que fornecem gráficos para análise do processo de assimilação. Até o presente momento, a plataforma fornece mapas com a geolocalização das observações, histogramas e séries temporais. A seguir uma breve descrição das funções disponíveis para análise dos dados, até então aplicadas apenas aos dados convencionais na versão V1.2.3 do **readDiag**.

- **plot**: fornece um mapa com pontos que indicam a geolocalização de obtenção da observação. Nesta função o usuário deve especificar a variável `varName` (por exemplo, `varName = 'uv'` para vento), o tipo `varType` (por exemplo, `varType = 281` para dados de estação a superfície com pressão) e o parâmetro que deseja visualizar na distribuição da barra de cores (*colorbars*).
- **ptmap**: assim como a função `plot`, esta função gera um mapa com a localização de todas as observações da variável `varName` sendo possível considerar mais de um tipo. O usuário pode definir `varType` com uma lista.
- **pvmmap**: de forma análoga a função `ptmap`, nesta função é possível visualizar para um específico `varType` a localização das observações considerando uma lista de variáveis `varName`.
- **pcount**: constrói um histograma da quantidade de dados para uma determinada variável considerando todos os tipos de dados disponíveis.

► **kxcount**: fornece um histograma da quantidade total de dados (soma do total de dados de todas as variáveis) para todos os tipos.

► **vcount**: fornece um histograma do total de dados para cada variável (considerando todos os tipos).

► **time_series**: esta função pode gerar até 6 séries temporais considerando as médias e desvios padrão dos valores de OmF e OmA, além do total de dados assimilados. A depender da quantidade de níveis determinada pelo usuário para análise, a função pode gerar diagramas (considerando a atmosfera dividida em níveis/camadas) ou gráficos de linha, quando considerada toda a atmosfera como uma única camada.

Exemplos de como utilizar as funções indicadas acima encontram-se nos notebooks disponibilizados na página de desenvolvimento do projeto no GitHub: <<https://github.com/GAD-DIMNT-CPTEC/readDiag>>.

2.2 Aspectos da avaliação da assimilação de dados de Radiância

O aprimoramento do processo de assimilação requer constante avaliação e análise dos arquivos de diagnóstico com o objetivo de verificar possíveis erros, problemas de modelagem e impacto da inserção de novos dados de satélites na PNT, buscando averiguar quais dados estão sendo assimilados e/ou rejeitados e as possíveis causas da rejeição.

A inserção dos dados de radiância no processo de assimilação, requer modificações no **readDiag** para acompanhamento e avaliação da assimilação destes dados. É importante considerar que o GSI fornece dois arquivos de diagnóstico, o primeiro contendo as informações de diagnóstico antes de iniciar o primeiro loop externo (1º outer loop) e o segundo contendo os dados obtidos após o segundo e último loop externo (2º outer loop) do processo de minimização executado pelo GSI (HU et al., 2016). O nome dos arquivos de diagnóstico de radiância são compostos por grafias padrões. Por exemplo:

`diag_amsua_n19_01.2024020100` e `diag_amsua_n19_03.2024020100`.

► **diag**: indica arquivo diagnóstico;

► **amsua**: nome do sensor de radiância (tipo de observação), no exemplo, o sensor é o AMSU-A (sensor de microondas *Advanced Microwave Sounding Unit-A*).

► **n19**: identifica o satélite cujo o sensor está acoplado, no exemplo a grafia n19

identifica o satélite NOAA-19.

- ▶ 01: indica que o arquivo é composto por dados de diagnóstico anterior ao 1º outer loop;
- ▶ 03: indica que o arquivo é composto por dados de diagnóstico obtidos após o 2º outer loop;
- ▶ 2024020100: identifica ano (2024), mês (02), dia (01) e hora UTC (00), nesta ordem.

Os arquivos mencionados fornecem dados como: identificação do sensor, identificação do satélite, latitude, longitude, elevação, canal, tempo, flag de análise de uso (*iuse*), índice de controle de qualidade (*idqc*), erros, observação (*obs*), observação - *background* (*OmF*), observação - análise (*OmA*), impacto, emissividade e geometria.

Com o objetivo de aproveitar a estrutura prévia criada no **readDiag** para leitura dos dados convencionais, o sensor de radiância será identificado através da variável *varName* e o satélite pela variável *varType*. Outra especificação importante para o tratamento dos dados é a escolha dos canais, cuja a coluna na tabela de dados é nomeada por *nchan*. As colunas *iuse* e *idqc* são importantes para definir quais observações foram assimiladas ou rejeitadas, sendo essenciais para analisar o impacto da assimilação de dados de radiância através das variáveis *OmF* e *OmA*.

Como determinar quais observações de radiância foram assimiladas? Para responder essa pergunta é necessário entender quais as flags e como essas flags são atribuídas as variáveis *iuse* e *idqc*.

Nos manuais do GSI (HU et al., 2016; HU et al., 2018) é possível encontrar as tabelas com as flags e seus significados para as variáveis *iuse* e *idqc*. A primeira, é utilizada para classificar dados convencionais e de radiância conforme a Tabela 2.1. Ou seja, uma observação foi possivelmente assimilada se seu valor de *iuse* for maior ou igual a 1.

Tabela 2.1 - Descrição das flags `iuse`.

<code>iuse</code>	descrição
-2	Não use.
-1	Monitorar se os diagnósticos foram produzidos.
0	Monitorar e usar somente em QC (controle de qualidade).
1	Usar dados com correção de viés completa.
2	Usar dados sem correção de viés de massa de ar.
3	Usar dados sem correção de viés dependente de ângulo.
4	Usar dados sem correção de viés.

Observe que os casos -1 e 0 são monitorados e -2 não é utilizado, o que indica que para uma determinada observação ser considerada rejeitada também é necessário que seu `iuse` seja 1. O que diferencia uma observação assimilada de uma observação rejeitada é a variável `idqc` que é resultado da rotina de controle de qualidade (script `qcmof.f90`). A depender da versão do GSI novas flags podem ser atribuídas ao `idqc`. Na versão utilizada neste trabalho e disponível na branch de desenvolvimento <https://projetos.cptec.inpe.br/projects/gsi/repository/entry/branch/gsi_t11824/src/gsi/qcmof.f90> os valores de `idqc` de zero à 12 são fixos para todos os sensores de radiância. As flags com valores superiores ou iguais a 50 indica avaliações específicas de cada sensor. Na Figura 2.1 há um recorte de uma tabela inserida em uma rotina do **readDiag** como comentário para visualização do significado das flags de `idqc` considerando os sensores AMSU-A e HIRS/4 (*High resolution Infrared Radiation Sounder*). Note que apenas o caso em que `idqc = 0` a observação é considerada boa, o que indica que para uma observação ser assimilada faz-se necessário,

```
iuse >= 1 & idqc == 0 .
```

idqc	flag	idqc	flag
0	Good Observations		SENSOR AMSUA QC_AMSUA
1	Reject due to flag in radinfo in setuprad	50	Reject because 'factch6 > limit' in subroutine qc_amsua
2	Failure in CRTM in setuprad	51	Reject because 'factch4 > limit' in subroutine qc_amsua
3	Reject due to gross check failure in setuprad	52	Reject because 'sval > limit' in subroutine qc_amsua over open water
4	Reject due to interchannel check (if one channel fails in group whole group thrown out)	53	Reject because 'factch5 > limit' in subroutine qc_amsua over open water
5	Reject due to not using over this surface in qc routine		SENSOR HIRS/4 QC_IRSND
6	Reject due to gross check in specific qc routine	50	Reject because 'wavenumber > 2400' in subroutine qc_irsnd
7	Reject due to 'cloud > limit' for channel in qc routine	51	Reject because 'wavenumber > 2000' in subroutine qc_irsnd
8	Reject due to inaccurate emissivity/surface temperature estimate in qc routine	52	Reject because goes sounder and satellite 'zenith angle > 60' in subroutine qc_irsnd
9	Reject due to observations being out of range in qc routine	53	Reject because of surface emissivity/temperature influence in subroutine qc_irsnd
11	Reject because outside the range of lsingleradob		
12	Reject due to cold-air outbreak area check in setuprad		

Figura 2.1 - Tabela com as flags do idqc de 0 à 12 e as flags específicas dos sensores AMSU-A e HIRS/4.

Na Figura 2.2 é mostrado o esquema de classificação proposto para determinar se uma observação é assimilada, rejeitada ou monitorada.

	idqc	iuse
Assimilated	== 0	>= 1
assimilated	== 0	>= -1 and < 1
Monitored		
rejected	!= 0	>= -1 and < 1
Rejected	!= 0	>= 1

Figura 2.2 - Esquema de classificação das observações de radiância com iuse e idqc.

Observe na Figura 2.2 que o esquema considera dois casos de dados monitorados:

- *monitorado-assimilado*: a observação é monitorada, porém, em relação ao idqc essa observação poderia ser assimilada;
- *monitorado-rejeitado*: a observação é monitorada, porém, em relação ao idqc essa observação seria rejeitada (falhou em algum teste da rotina de controle de qualidade).

Os aspectos mencionados acima são muito importantes para averiguar o desempenho da assimilação de dados de radiância e seu impacto na correção de viés, ou seja, nos valores de OmF e OmA. Além disso, saber identificar os dados que foram assimilados pelo GSI permite determinar a densidade de observações que contribuíram para a PNT e seu impacto. Determinar os canais que obtiveram observações rejeitadas e identificar o motivo desta rejeição permite sugerir apontamentos para melhorar a assimilação e consequentemente a PNT.

No próximo capítulo são apresentadas as funções criadas e modificadas nos scripts do **readDiag** para visualização, classificação das observações e análise estatística dos valores de OmF e OmA.

3 Assimilação de dados de Radiância: implementação

O objetivo deste capítulo é descrever o que foi implementado e modificado na estrutura “`__main__.py`” para leitura e análise estatística dos dados de diagnóstico da radiância. Uma das funcionalidades do **readDiag** que permite analisar o impacto das observações assimiladas é a função **time_series** desenvolvido para gerar séries temporais dos valores médios e de desvio padrão dos parâmetros OmF e OmA, além de fornecer uma série temporal da quantidade de observações assimiladas de dados convencionais. Adaptar esta função para obter uma série temporal dos dados de radiância requer muitas modificações o que tornaria o script da função **time_series** grande e mais complexo. Sendo assim, optou-se por elaborar uma função própria que forneça séries temporais de OmF e OmA das observações de radiância: a função “**time_series_radi**”.

3.1 Função **time_series_radi**

Nesta seção, apresenta-se as modificações necessárias para composição da função “**time_series_radi**”, que será responsável por oferecer gráficos para a análise temporal dos dados de radiância obtidos inicialmente pelo sensor de microondas *Advanced Microwave Sounding Unit-A* (AMSU-A) nos satélites NOAA-15, NOAA-18, NOAA-19 e MetOp-B. Este sensor foi inicialmente considerado pois possui dados disponíveis em todos os experimentos à disposição para avaliação da assimilação.

Com o objetivo de obter uma série temporal para diferentes canais do sensor, a função “**time_series**” foi formulada para elaborar *digramas de hovmoller* das médias globais e desvios padrão de OmF e OmA. A construção da função “**time_series_radi**” no **readDiag** foi realizada considerando a função “**time_series**” elaborada para a construção dos *digramas de hovmoller* de dados convencionais.

Algumas diferenças importantes são consideradas. O *digrama de hovmoller* da série temporal de dados convencionais foi pensado com o objetivo de representar as médias globais dos parâmetros OmF e OmA dos sensores identificados com a variável `varName`: pressão (*ps*), componentes do vento horizontal (*uv*), temperatura (*T*), umidade específica (*q*) e dados de radio ocultação GNSS como refratividade ou ângulo de ocultação (*gps*). Para os dados de radiância a variável considerada recebe o nome do sensor acoplado nos satélites mencionados, AMSU-A, sendo necessário considerar `varName = amsua`. No caso dos dados convencionais, a construção do diagrama é realizada considerando as médias globais dos parâmetros em um intervalo de tempo e em diferentes níveis (`Levels`). Para cada nível é determinada a

série temporal que será representada em uma linha do diagrama. **Considerando que o objetivo é analisar a série temporal dos dados de radiância e para esses não há um nível específico, mas a distribuição vertical dos canais fornecida por uma função peso, expressa por uma distribuição normal, com seu máximo em um determinado nível, essa avaliação é feita em cada canal em que o sensor trabalha.** Assim, cada linha do *diagrama de hovmoller* deve representar a série temporal das médias globais de OmF ou OmA em um determinado canal. Dessa forma, como no caso convencional, a função “time_series_radi” deve retornar seis diagramas. Para cada parâmetro, OmF e OmA, respectivamente, são construídos os diagramas:

- I - das médias globais em função do tempo;
- II - dos desvios padrão em função do tempo;
- III - da quantidade de dados assimilados em função do tempo.

A função “time_series_radi” apresenta os seguintes argumentos de entrada, com um exemplo e sua respectiva descrição:

```
varName = "amsua"      # Radiance Sensor
varType = "n15"        # Source Type satellite
mask    = None         # Mask the data by chosen used/not used data,
                        # ex: mask="iuse>=1 & idqc==0"
dateIni  = 2013010100  # Inicial Date
dateFin  = 2013010900  # Final Date
nHour    = "06"        # Time Interval
vminOMA  = -4.0        # Y-axis Minimum Value for OmF or OmA
vmaxOMA  = 4.0         # Y-axis Maximum Value for OmF or OmA
vminSTD  = 0.0         # Y-axis Minimum Value for Standard Deviation
vmaxSTD  = 14.0        # Y-axis Maximum Value for Standard Deviation
channel  = 1           # Time Series channel, if any (None),
                        # all nchan are plotted
```

Note que a quantidade de canais presente na análise é determinado pelo argumento channel. Se para channel for atribuído a tag de um único canal, então a função retorna gráficos de linha das médias de OmF e OmA, do desvio padrão e da quantidade de dados. Agora se para channel for atribuído uma lista de tags de canais ou a expressão None, então, a função retorna os 6 *diagramas de hovmoller* com as estatísticas dos canais listados ou, no caso None, de todos os canais do sensor.

Para observar exemplos de como utilizar a função “time_series_radi” o leitor pode

consultar o notebook gerado para análise dos dados de radiância e disponibilizado no GitHub do projeto: <https://github.com/GAD-DIMNT-CPTEC/readDiag/blob/master/notebooks/readDiag_Radiancia_tutorial_completo-pt_br.ipynb> e no Apêndice A deste relatório. A Figura 3.1 apresenta o *diagrama de hovmoller* das médias de OmA das observações assimiladas nos 15 canais do sensor AMSU-A, satélite NOAA-19, no período de 01/03/2024 à 31/03/2024.

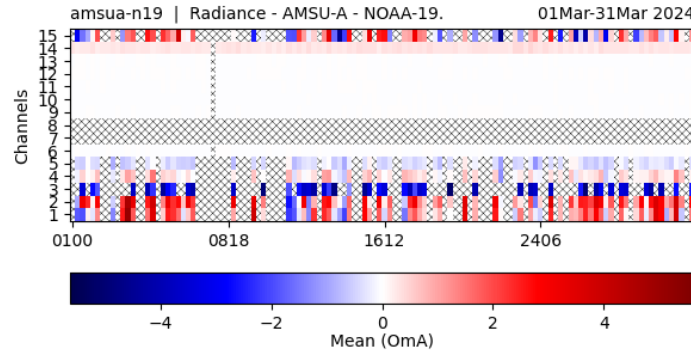


Figura 3.1 - Exemplo de *diagrama de hovmoller* obtido através da função `time_series_radi` do `readDiag`.

Note que no diagrama da Figura 3.1 aparecem regiões rachuradas que indicam ausência de dados assimilados. Observe que os canais 7 e 8 não obtiveram dados assimilados em toda a série. Tal fato leva a investigar o status desses dados. Esses canais são monitorados? ou seja, `iuse == -1`. Ou ainda, tais canais obtiveram todas as observações rejeitadas? `idqc != 0`?

Para suprir a necessidade de verificar e visualizar a classificação das observações de radiância em assimilado, monitorado ou rejeitado que foram realizadas modificações na rotina da função “`statcount`” do `readDiag`, visando obter gráficos que auxiliem no processo de visualizar geograficamente a posição dessas observações e sua classificação, além de fornecer uma série temporal com a quantidade de dados assimilados, monitorados e rejeitados do canal indicado. Uma breve descrição do que foi desenvolvido na função “`statcount`” é mencionado na próxima seção.

3.2 Função `statcount`

Inicialmente a função “`statcount`” foi idealizada para fornecer uma série temporal de mapas com pontos indicando a localização das observações de uma determinada variável `varName` e tipo `varType` de dados convencionais onde as cores classificam

essas observações em assimiladas, monitoradas e rejeitadas. Além da série temporal de mapas, a função também fornece um gráfico com a série temporal do total de dados assimilados, monitorados e rejeitados. Para obter a classificação dos dados de radiância, foi introduzido um novo argumento na função (`channel`) de forma que se `channel = None` o código executa a classificação das observações convencionais e se `channel = n° canal` é executado a classificação das observações de radiância.

A classificação das observações de radiância é apresentada na Figura 2.2. Na Figura 3.2 o leitor pode observar um exemplo de mapa com a geolocalização e classificação das observações obtidas pelo sensor AMSU-A, no satélite NOAA-19, canal 06 e na Figura 3.3 referente ao canal 08. Um exemplo da série temporal da quantidade total de dados assimilados, monitorados e rejeitados pode ser observado na Figura 3.4.

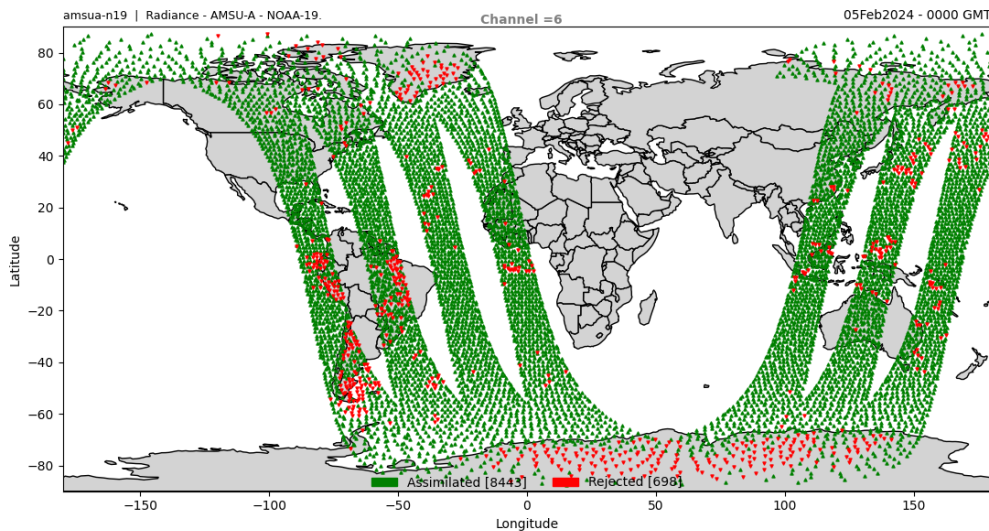


Figura 3.2 - Exemplo de mapa gerado pela função `statcount` do `readDiag`, identificando os dados do canal 06 em assimilados (verde) e rejeitados (vermelho).

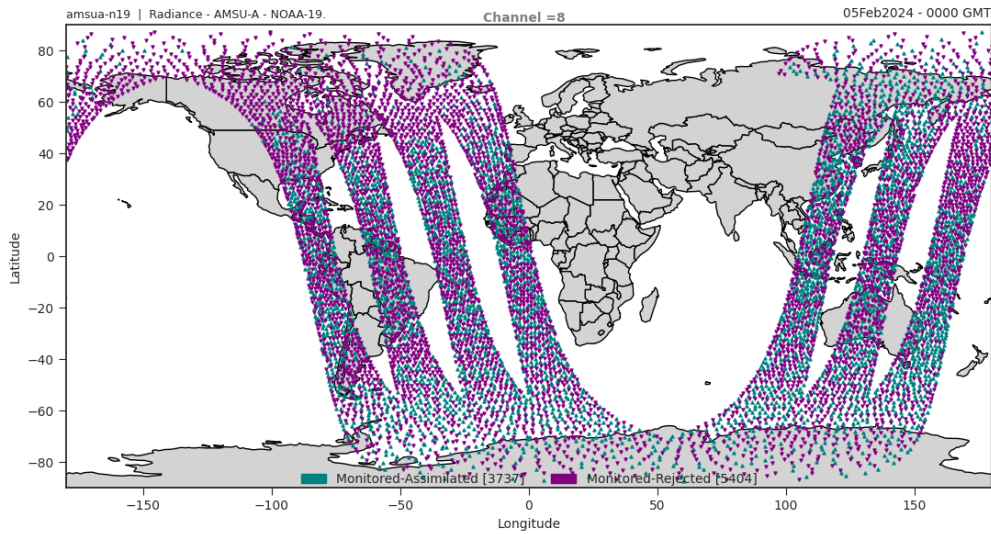


Figura 3.3 - Exemplo de mapa gerado pela função `statcount` do `readDiag`, identificando os dados do canal 08 em monitorados-assimilados (teal) e monitorados-rejeitados (purple).

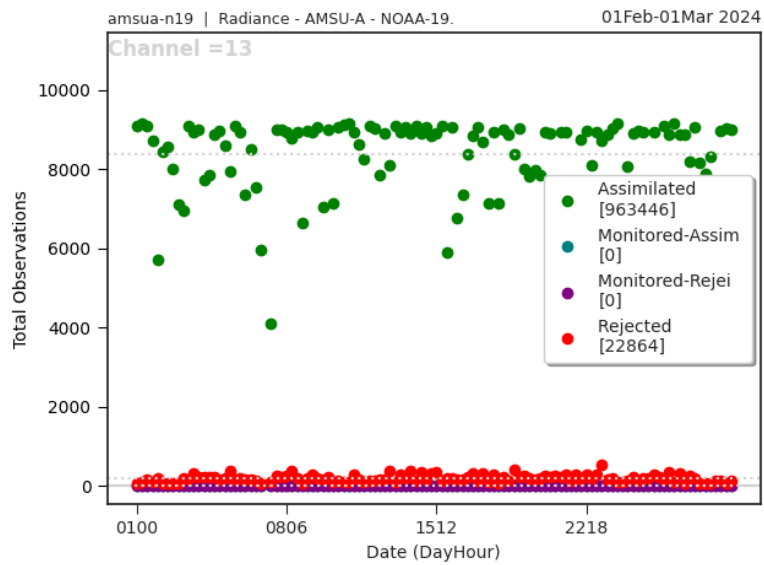


Figura 3.4 - Exemplo da série temporal do total de dados assimilados, monitorados e rejeitados, gerado pela função `statcount` do `readDiag` para dados de radiância.

A função `statcount` serve como um complemento para entender o que está acontecendo nos canais cuja a região do diagrama está rachurada, além de permitir identificar quais observações foram rejeitadas.

3.3 Função plot

Outra função que foi modificada no **readDiag** é a função **plot**. Esta função fornece ao usuário um mapa com a geolocalização das observações e determina a cor do marcador da observação conforme o valor de um parâmetro identificado pelo argumento **param**, gerando uma barra de cores (colorbar) definida conforme os valores de mínimo e máximo do parâmetro.

De modo geral, não foram necessárias modificações para a plotagem das observações de radiância. As alterações realizadas tiveram o objetivo de inserir os argumentos **minVal = None** e **maxVal = None**, de modo que o usuário possa fixar os valores de mínimo e máximo do colorbar, permitindo melhor visualização dos resultados.

Para gerar os resultados de radiância nesta função é importante considerar no termo **mask** o **iuse** e o **idqc**, uma vez que, diferente dos dados convencionais, o valor do **idqc** é essencial para identificar se a observação foi assimilada ou não. Na Figura 3.5 é apresentado um exemplo do mapa gerado pela função **plot** considerando **param = 'oma'**, **minVal = -10**, **maxVal = 10** e **mask =**

```
"(nchan==" + str(channel) + ") & (iuse >= 1 & idqc == 0)".
```

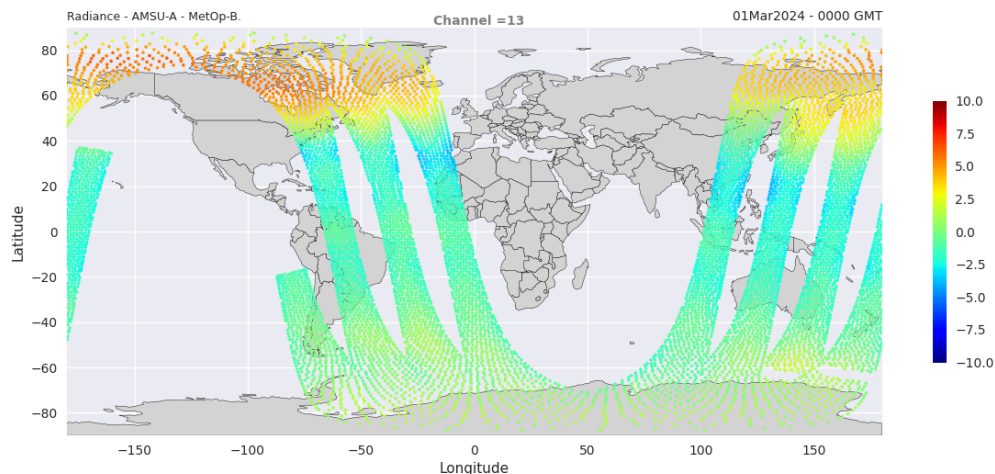


Figura 3.5 - Exemplo de mapa gerado pela função **plot** do **readDiag**, identificando os dados do canal 13 em relação ao parâmetro **OmA** e fixando os valores de mínimo e máximo do intervalo de cores.

Desta forma, encerra-se as principais alterações realizadas no **readDiag** para leitura e análise dos dados de diagnóstico da radiância. As modificações descritas neste capítulo foram desenvolvidas em uma *branch* de um *fork* da versão *main* do projeto no GitHub, versão 1.2.3 disponível em <<https://github.com/GAD-DIMNT-CPTEC/readDiag/releases/tag/v1.2.3>>. As modificações foram analisadas por um terceiro membro do projeto e inseridas (*merge*) em uma nova versão *main* do projeto: versão 1.3.2 disponível em <<https://github.com/GAD-DIMNT-CPTEC/readDiag/releases/tag/v1.3.2>>.

No próximo capítulo, apresenta-se resultados obtidos através das ferramentas adaptadas para análise dos resultados de radiância gerados pela versão em desenvolvimento do SMNA.

4 Resultados: análise dos dados de radiância

Nas seções seguintes, serão realizadas análises da assimilação de dados de radiância do sensor AMSU-A nos experimentos 19 (**EXP19**), 20 (**EXP20**) e 21 (**EXP21**) descritos na Tabela 4.1. A ideia central é investigar através das séries temporais e distribuições espaciais, melhorias na assimilação de dados de radiância entre as versões V2.4.0 (em desenvolvimento) e V2.3.1 (versão modelo de controle ([SAPUCCI et al., 2024](#))) do SMNA (*Sistema de Modelagem Numérica e Assimilação de dados*).

Tabela 4.1 - Lista de experimentos realizados para avaliar a versão SMNA V2.4.0.

Label	Descrição	Período
EXP19	SMG V2.3.1: Rodada pré operacional da versão SMNA 2.3.1 sendo realizada na DIPTC no CX50, sob os cuidados do IO. Repositório: http://ftp1.cptec.inpe.br/pesquisa/das/EXPS_-SMNA/EXP19/	2024020100- 2024033118
EXP20	SMNA V2.4.0: Rodada teste na EGEON da versão SMNA 2.4.0 que conta com a versão do GSI 3.7, sob os cuidados de José Aravequia. Repositório: http://ftp1.cptec.inpe.br/pesquisa/das/EXPS_-SMNA/EXP20/	2024020100- 2024022918
EXP21	SMNA V2.4.0: Rodada teste na EGEON da versão SMNA 2.4.0 a mesma que o EXP20 mas com correção de vies: http://ftp1.cptec.inpe.br/pesquisa/das/EXPS_-SMNA/EXP21/	2024030100- 2024033118

Observe que para analisar os resultados e comparar as versões do SMNA é necessário realizar dois tipos de comparações: a primeira entre os experimentos **EXP19** e **EXP20**, compreendendo o período de 01/02/2024 à 29/02/2024 e a segunda entre os experimentos **EXP19** e **EXP21** no período de 01/03/2024 à 31/03/2024.

4.1 Comparação entre os **EXP19** e **EXP20**

Como mencionado a comparação dos experimentos **EXP19** e **EXP20** compreende o período: 01/02/2024 à 29/02/2024. Pretende-se averiguar o status da versão em desenvolvimento do SMNA, se as modificações realizadas até então resultaram em

melhorias na assimilação de dados de radiância e indicar possíveis ajustes.

Inicialmente, observa-se os resultados referente a série temporal de OmF e OmA das observações assimiladas e coletadas pelo satélite NOAA-15 e disponíveis no **EXP19**. Como pode ser observado na Figura 4.1 a coluna de gráficos à esquerda apresenta as séries temporais para o parâmetro OmF e a coluna de gráficos à direita os resultados referente ao parâmetro OmA. Verifica-se que os canais 7 à 10, 12 e 13 são assimilados em todo o período com uma quantidade de observações assimiladas superior a 4/5 mil. Nota-se que os canais 1 à 5 e 15 obtiveram uma quantidade pequena de dados assimilados em um período curto da série temporal, entre os dias 13 e 20 de fevereiro. Comparando as médias globais, é possível observar que as médias de OmF nos canais 7 à 10, 12 e 13 são ligeiramente superiores as médias globais de OmA em todo o período da série o que indica uma melhora do sistema de assimilação, uma vez que o erro observação - análise é menor.

Para verificar a localização espacial das observações assimiladas no período de 13 à 20 de fevereiro nos canais de 1 à 5 e 15, observa-se a distribuição da variável impacto total (IT) no dia 19/02/2024 às 12:00 UTC. A Figura 4.2 mostra a localização geográfica das observações assimiladas. Constatado que poucos dados foram assimilados, uma segunda análise pode ser realizada ao verificar o que está acontecendo com as observações que não foram assimiladas. Todavia, existem duas possibilidades para esses dados: são classificados como rejeitados ou monitorados. Utilizando a função “statcount” da classe “plot_diag”, é possível verificar se as observações que não aparecem na Figura 4.2 são rejeitadas ou monitoradas. Na Figura 4.3 fica evidente que quase a totalidade dos dados são rejeitados, mostrando que apenas 12 observações foram assimiladas.

aqui temos que colocar o texto sempre no impessoal, estava 'vamos observar'

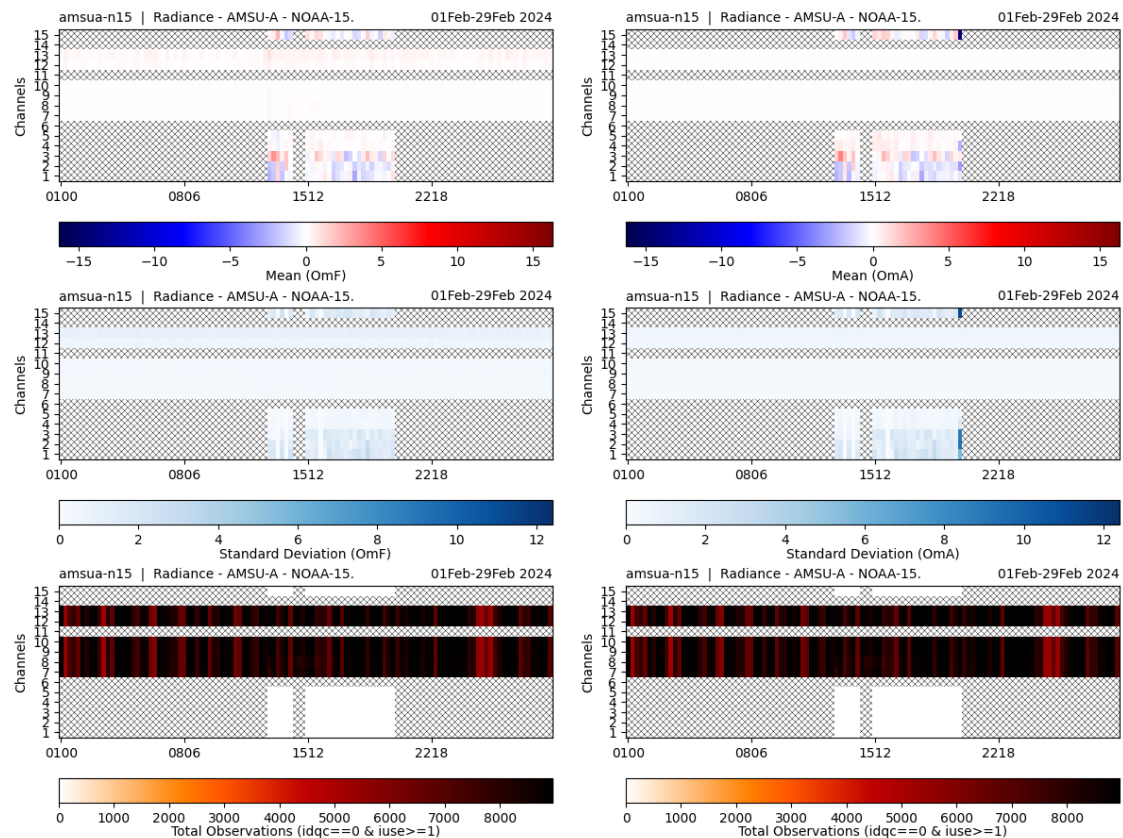


Figura 4.1 - Série temporal das médias globais (1ª linha), desvio padrão (2ª linha) e quantidade de dados assimilados (3ª linha) dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-15 no período 2024020100 - 2024022918.

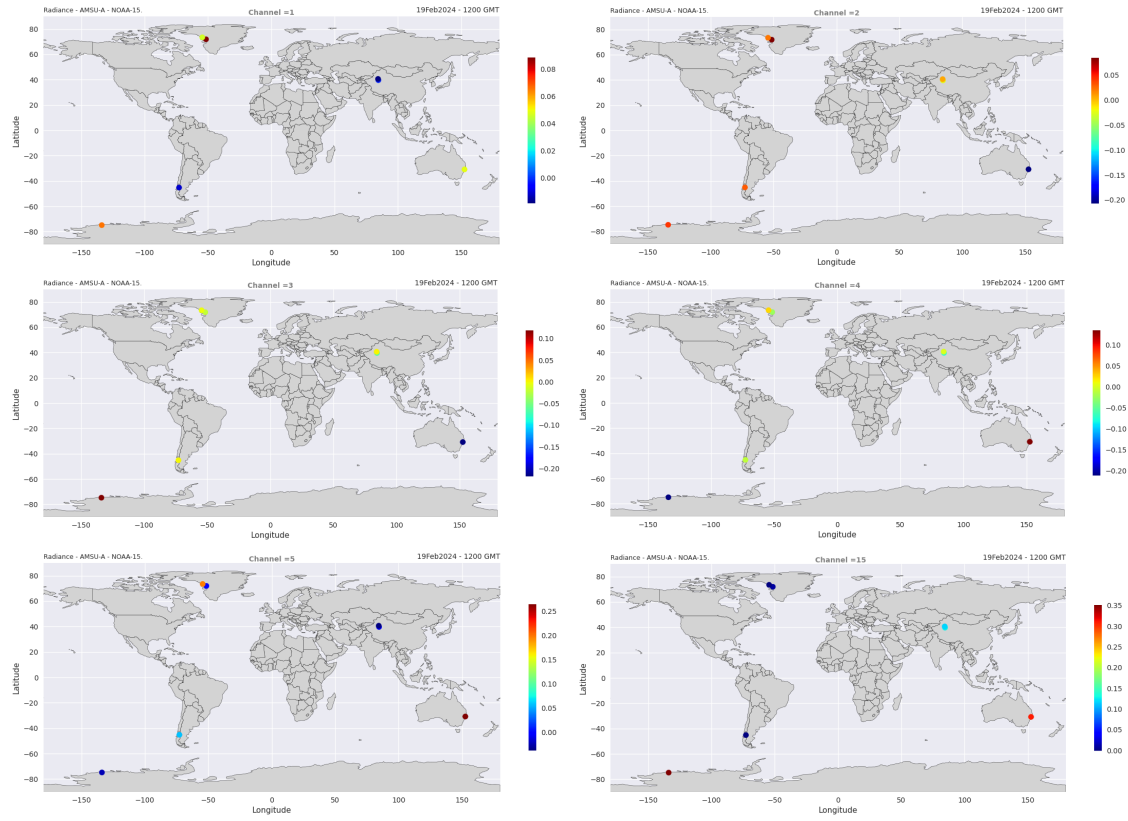


Figura 4.2 - Distribuição espacial das observações assimiladas do sensor AMSU-A no satélite NOAA-15, canais: 1, 2, 3, 4, 5 e 15, em 2024021912.

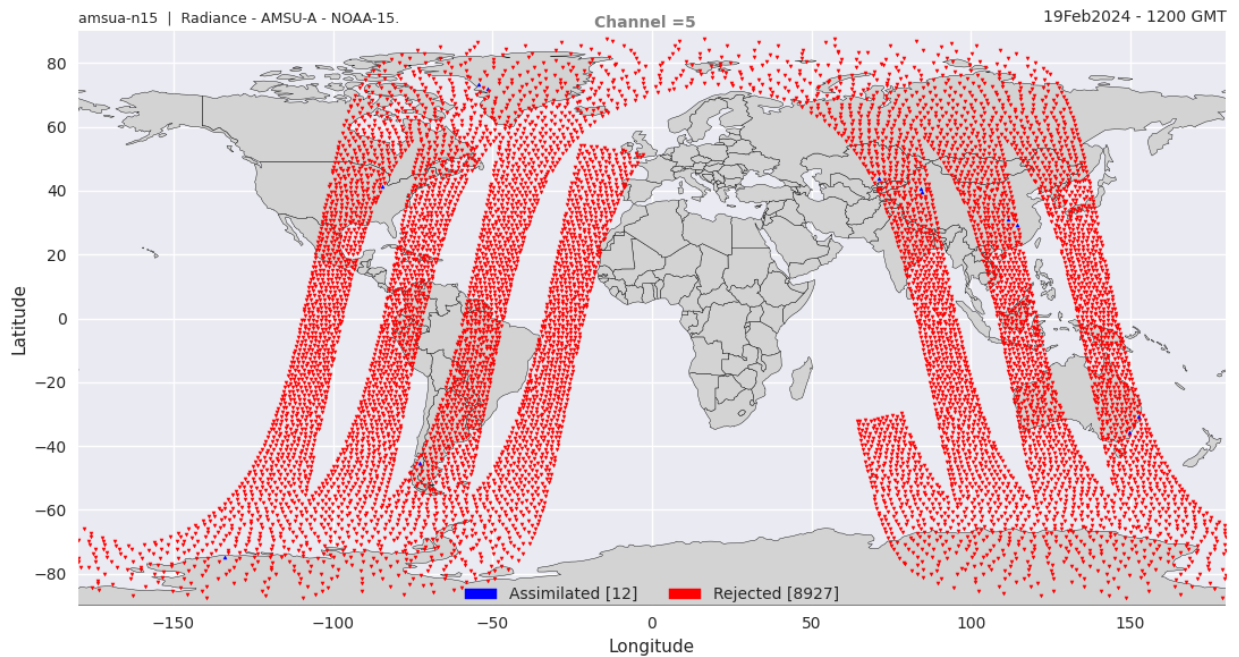


Figura 4.3 - Distribuição espacial das observações assimiladas e rejeitadas do sensor AMSU-A no satélite NOAA-15, canal 5, em 2024021912.

As Figuras 4.4 e 4.5 apresentam as séries temporais para todos os canais dos satélites NOAA-18 e NOAA-19 respectivamente (o satélite MetOp-b não consta na base de dados dos experimentos no mês de fevereiro de 2024). Observa-se que nos canais 6, 7, 10 à 14 do satélite NOAA-18 e 6, 9 à 14 do satélite NOAA-19 os valores de média e desvio padrão de OmA são menores que os valores de OmF. No dia 07/02/2024 às 06 UTC, observa-se uma pequena falha nas séries temporais do satélite NOAA-19 devido a falta de arquivos na base de dados do **EXP19**.

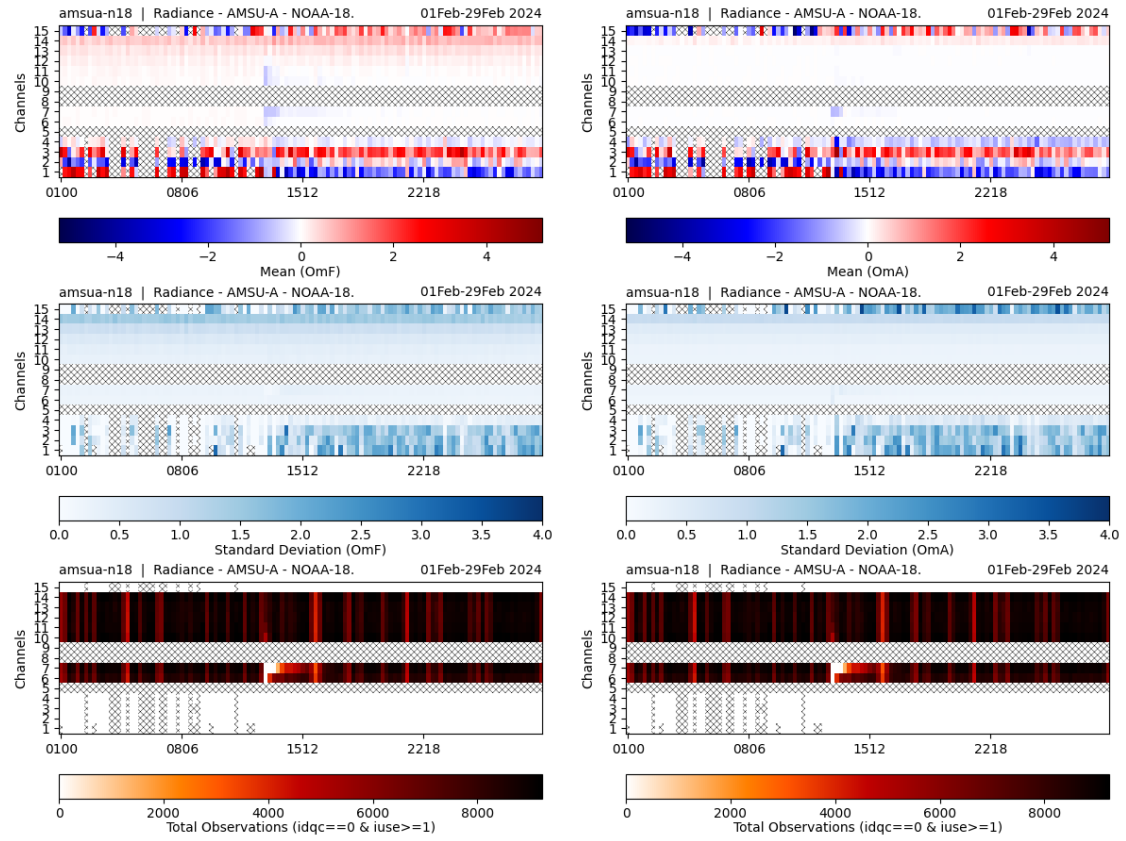


Figura 4.4 - Série temporal das médias globais (1ª linha), desvio padrão (2ª linha) e quantidade de dados assimilados (3ª linha) dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-18 no período 2024020100 - 2024022918.

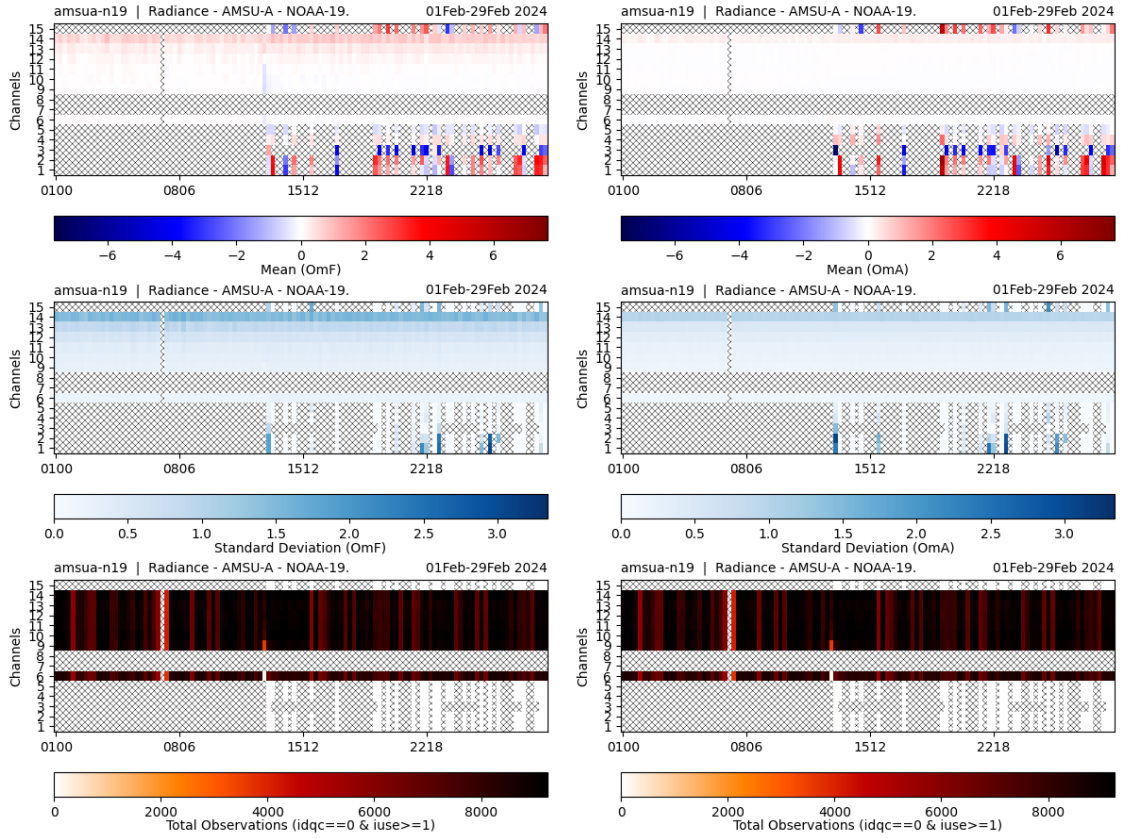
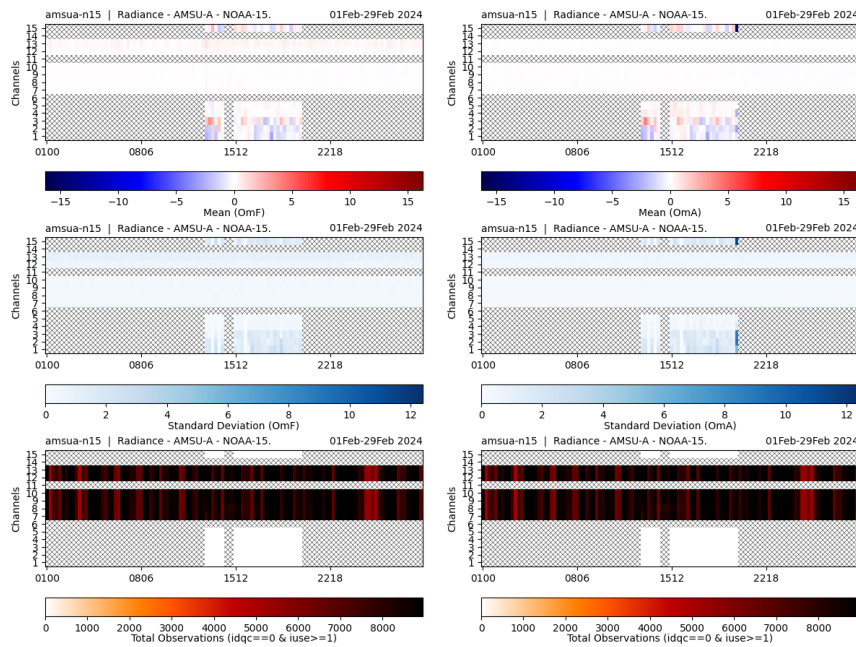


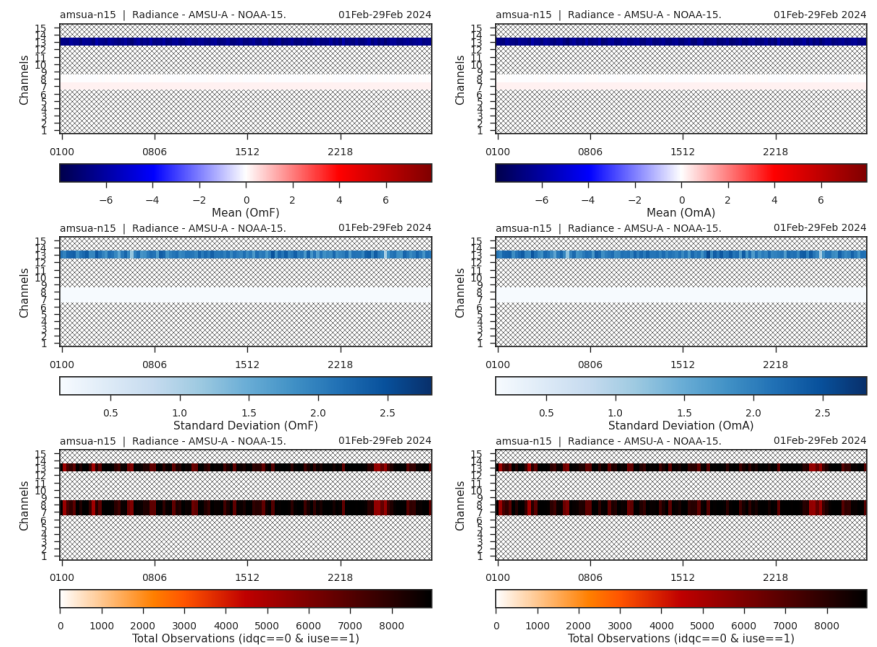
Figura 4.5 - Série temporal das médias globais (1ª linha), desvio padrão (2ª linha) e quantidade de dados assimilados (3ª linha) dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-19 no período 2024020100 - 2024022918.

A seguir apresenta-se os resultados do **EXP19** e **EXP20**, lado a lado, para averiguar possíveis diferenças e/ou melhorias na assimilação de dados. Os resultados dos dados de radiância providos dos Satélites NOAA-15-18 e 19 são observados nas Figuras 4.6, 4.7 e 4.8 respectivamente. Em uma breve comparação, é evidente que o experimento 19 apresenta uma quantidade maior de canais assimilados em quase toda a série temporal e em todos os satélites considerados, com uma predominância dos canais 6 à 13. No experimento 20, a quantidade de canais assimilados é baixa limitando-se a 3 ou 4 canais. Tal fato indica que faz-se necessário entender o que está acontecendo com os canais que não foram assimilados no **EXP20**. Entretanto, como indicado na Tabela 4.1, tal experimento foi executado com a versão em desenvolvimento do SMNA, porém, sem correção de viés.

Na sequência são apresentadas as séries temporais obtidas com os dados de março de 2024 dos experimentos 19 e 21.

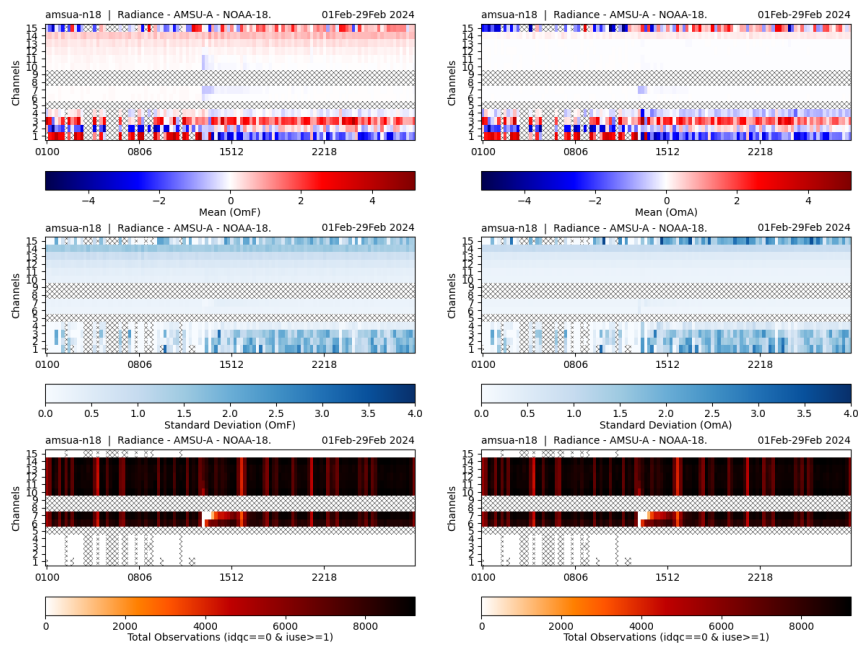


(a) EXP19

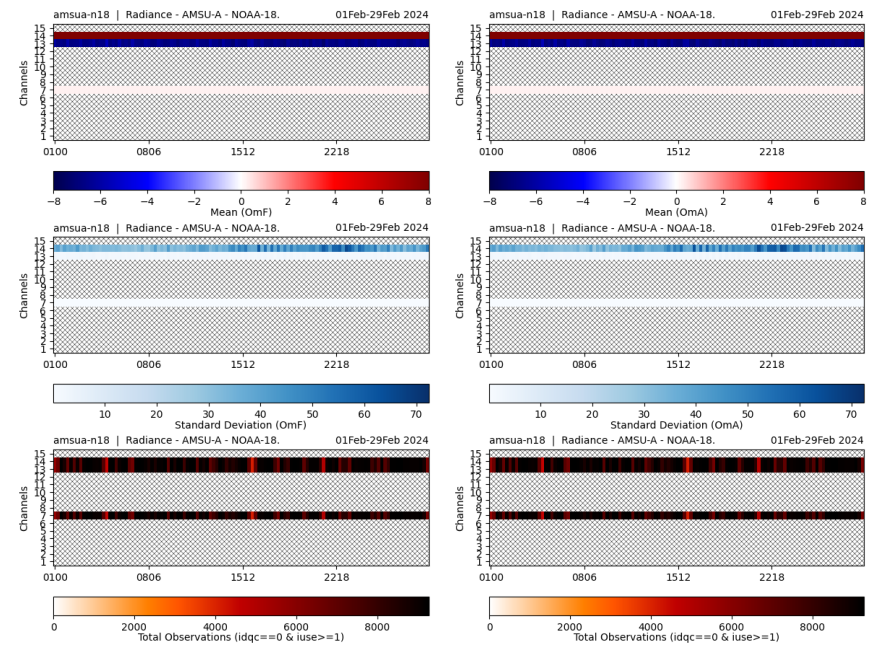


(b) EXP20

Figura 4.6 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-15 nos experimentos 19 (a) e 20 (b), período 2024020100 - 2024022918.

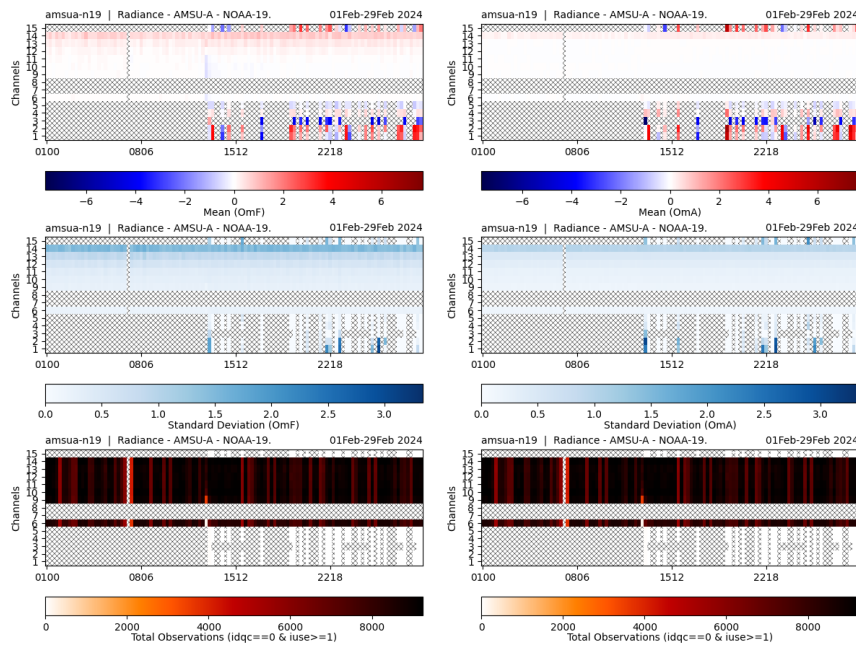


(a) **EXP19**

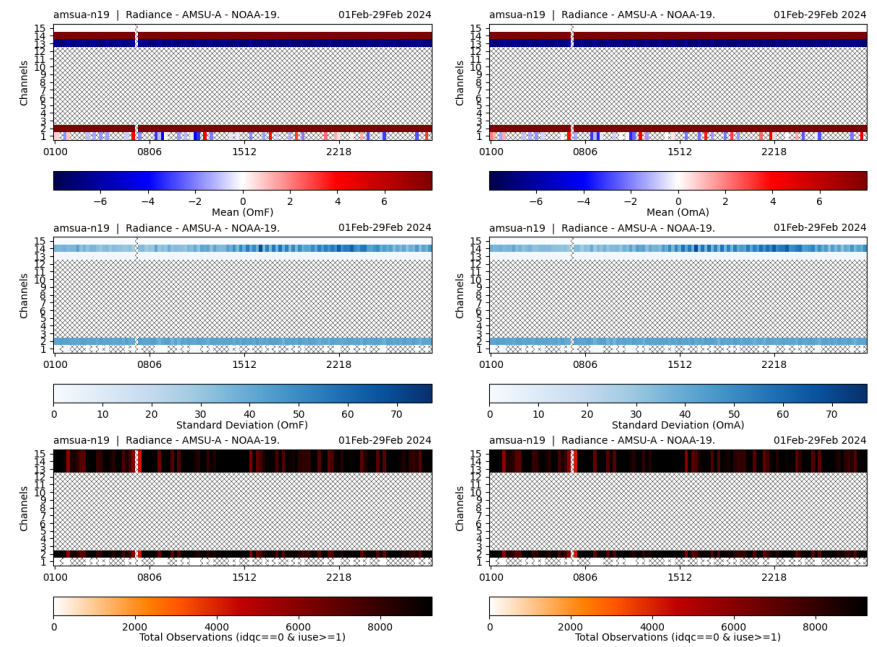


(b) **EXP20**

Figura 4.7 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-18 nos experimentos 19 (a) e 20 (b), período 2024020100 - 2024022918.



(a) EXP19



(b) EXP20

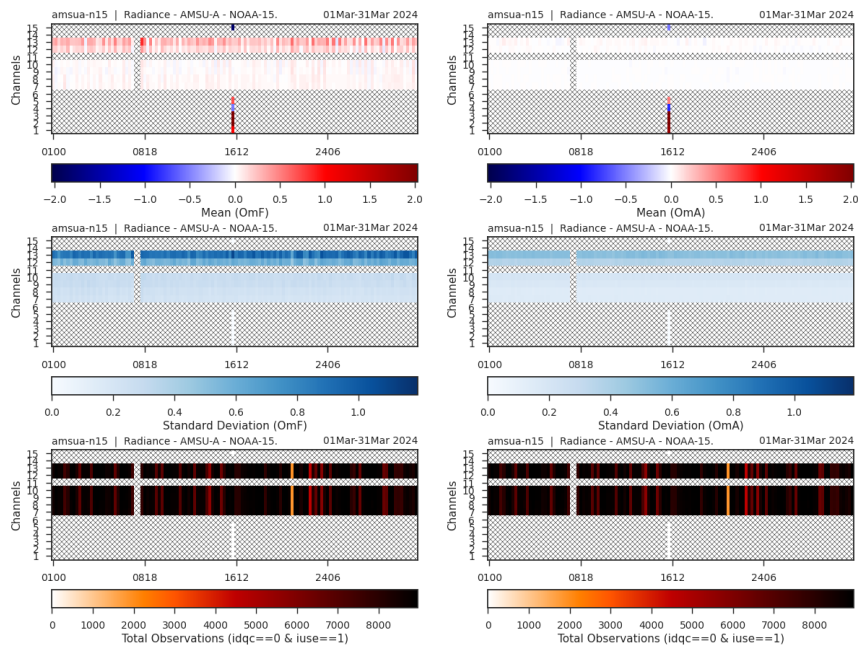
Figura 4.8 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-19 nos experimentos 19 (a) e 20 (b), período 2024020100 - 2024022918.

4.2 Comparação entre os EXP19 e EXP21

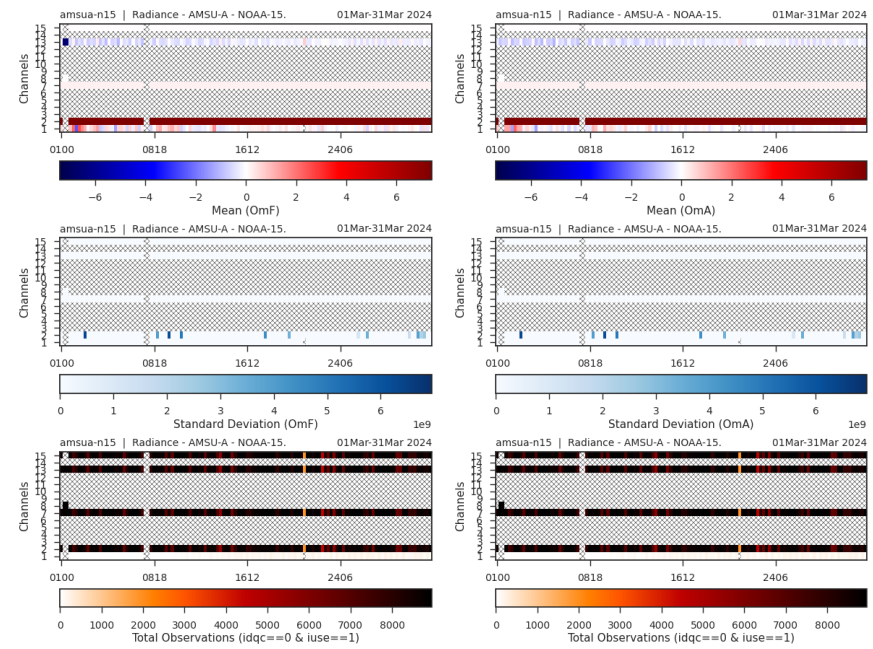
Como mencionado anteriormente a comparação dos experimentos 19 e 21 é realizada no período de 01/03/2024 à 31/03/2024, pois, é o período que coincide entre ambos os experimentos.

As Figuras 4.9, 4.10 e 4.11 mostram a série temporal das médias de OmF e OmA das observações assimiladas pelos satélites NOAA-15, 18 e 19 nos experimentos 19 (a) e 21 (b). Novamente, observa-se que no experimento 19 a quantidade de canais assimilados é superior a quantidade de canais assimilados no experimento 21, onde, neste último verifica-se que apenas 5 ou 6 canais são assimilados. Diferentemente dos resultados apresentados pelo **EXP19** os valores médios de OmA e OmF parecem coincidir, entretanto, mostram que os valores decrescem para zero no decorrer do tempo. Em relação aos desvios padrão, nota-se que os diagramas de OmA e OmF são muito similares.

Para visualizar melhor essa similaridade dos valores de OmF e OmA faz-se necessário especificar um canal e analisar a série temporal de ambos os parâmetros na forma de gráfico, o que será realizado na próxima seção.

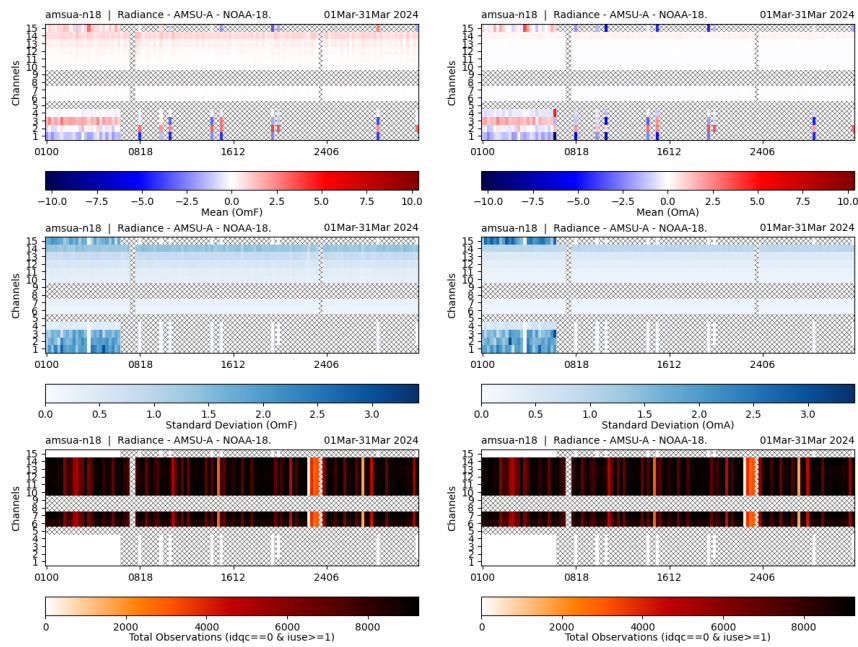


(a) **EXP19**

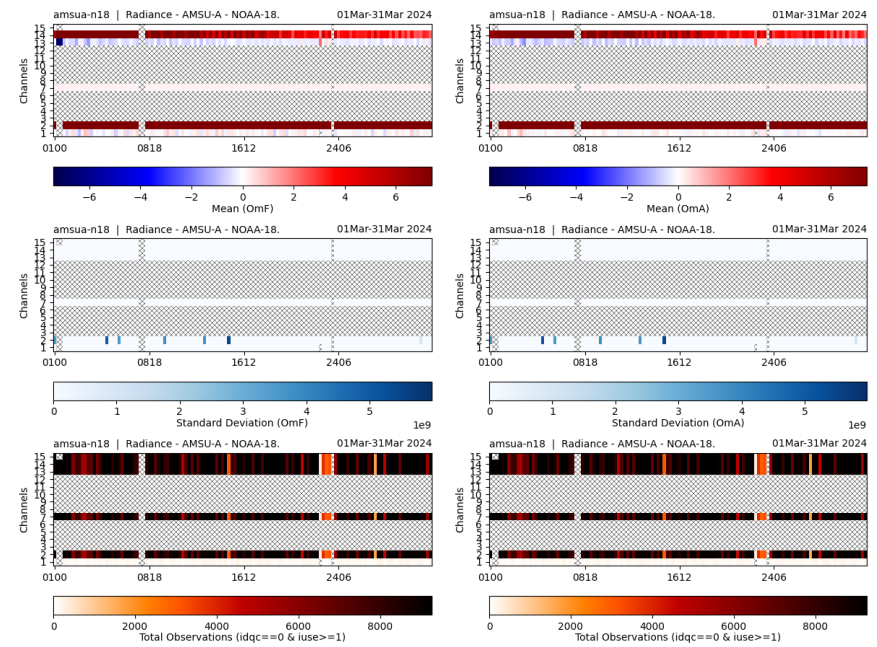


(b) **EXP21**

Figura 4.9 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-15 nos experimentos 19 (a) e 21 (b), período 2024030100 - 2024033118.

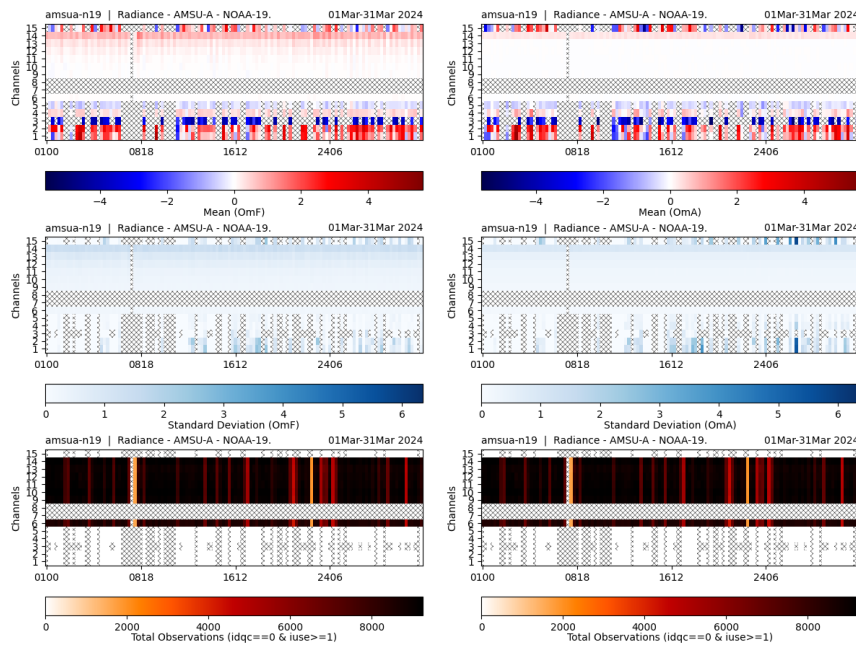


(a) EXP19

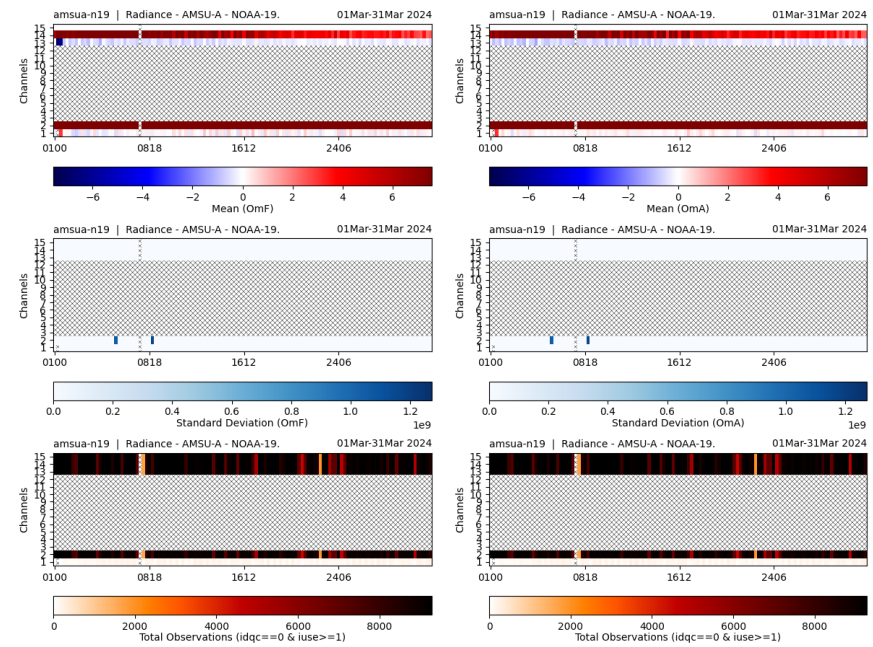


(b) EXP21

Figura 4.10 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-18 nos experimentos 19 (a) e 21 (b), período 2024030100 - 2024033118.



(a) EXP19



(b) EXP21

Figura 4.11 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite NOAA-19 nos experimentos 19 (a) e 21 (b), período 2024030100 - 2024033118.

4.3 Análise por canal

Nesta subseção são apresentados resultados mais pontuais, considerando especificamente um canal para análise. Os canais 13 e 14 são escolhidos, uma vez que, ao observar a Tabela 4.2, o canal 13 é predominantemente assimilado em todos os satélites (exceto MetOp-b) e experimentos, enquanto que, o canal 14 é assimilado nos satélites NOAA-18-19 e MetOp-b, e monitorado no satélite NOAA-15. A Tabela 4.2 foi construída observando os diagramas de séries temporais e analisando através da função “`statcount`” a classificação das observações e canais que não foram assimilados.

Na Tabela 4.2 a cor verde indica que o canal obteve mais do que 2 mil observações assimiladas em todo o período de estudo do experimento. A cor cinza indica que os dados do satélite no experimento não estão disponíveis. A cor vermelho indica que naquele canal a predominância é de observações rejeitadas e a cor azul identifica os canais que são monitorados. A ideia desta Tabela é sintetizar a análise dos experimentos, satélites e canais, de modo que facilite a compreensão dos resultados.

Tabela 4.2 - Classificação dos canais em todos os experimentos: assimilado, monitorado e rejeitado.

Satélite	NOAA-15			NOAA-18			NOAA-19			MetOp-b		
EXP	19	20	21	19	20	21	19	20	21	19	20	21
CH1	Rejeitado	Monitorado	Rejeitado	Rejeitado	Monitorado	Rejeitado	Rejeitado	Rejeitado	Rejeitado	None	Monitorado	Monitorado
CH2	Rejeitado	Monitorado	Assimilado	Rejeitado	Monitorado	Assimilado	Rejeitado	Assimilado	Assimilado	None	Monitorado	Monitorado
CH3	Rejeitado	Monitorado	Rejeitado	Rejeitado	Monitorado	Rejeitado	Rejeitado	Rejeitado	Rejeitado	None	Monitorado	Monitorado
CH4	Rejeitado	Monitorado	Rejeitado	Rejeitado	Monitorado	Rejeitado	Rejeitado	Rejeitado	Rejeitado	None	Monitorado	Monitorado
CH5	Rejeitado	Monitorado	Rejeitado	Monitorado	Monitorado	Monitorado	Rejeitado	Rejeitado	Rejeitado	None	Monitorado	Monitorado
CH6	Monitorado	Monitorado	Monitorado	Assimilado	Rejeitado	Rejeitado	Assimilado	Rejeitado	Rejeitado	None	Monitorado	Monitorado
CH7	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Monitorado	Monitorado	Monitorado	None	Monitorado	Monitorado
CH8	Assimilado	Assimilado	Rejeitado	Monitorado	Monitorado	Monitorado	Monitorado	Monitorado	Monitorado	None	Assimilado	Rejeitado
CH9	Assimilado	Rejeitado	Rejeitado	Monitorado	Monitorado	Monitorado	Assimilado	Rejeitado	Rejeitado	None	Rejeitado	Rejeitado
CH10	Assimilado	Rejeitado	Rejeitado	Assimilado	Rejeitado	Rejeitado	Assimilado	Rejeitado	Rejeitado	None	Rejeitado	Rejeitado
CH11	Monitorado	Monitorado	Monitorado	Assimilado	Rejeitado	Rejeitado	Assimilado	Rejeitado	Rejeitado	None	Rejeitado	Rejeitado
CH12	Assimilado	Rejeitado	Rejeitado	Assimilado	Rejeitado	Rejeitado	Assimilado	Rejeitado	Rejeitado	None	Rejeitado	Rejeitado
CH13	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	None	Assimilado	Assimilado
CH14	Monitorado	Monitorado	Monitorado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	Assimilado	None	Assimilado	Assimilado
CH15	Rejeitado	Monitorado	Assimilado	Rejeitado	Monitorado	Assimilado	Rejeitado	Assimilado	Assimilado	None	Monitorado	Monitorado

Assimilado

Monitorado

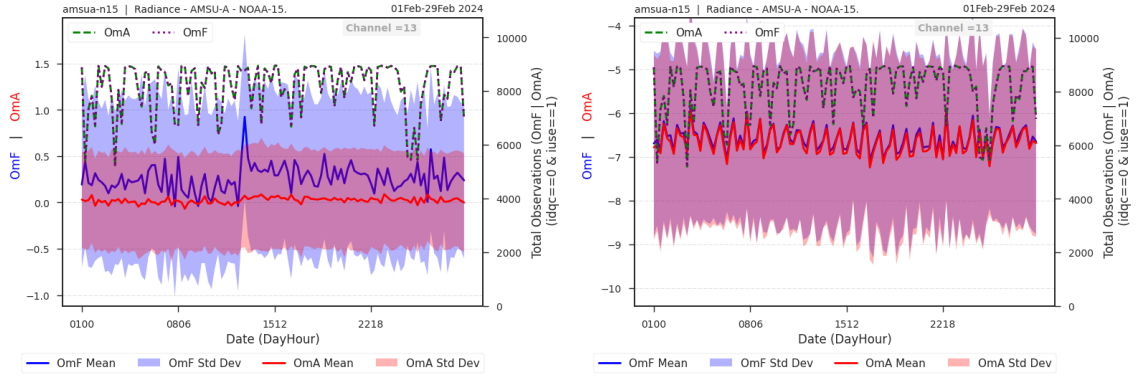
Rejeitado

None

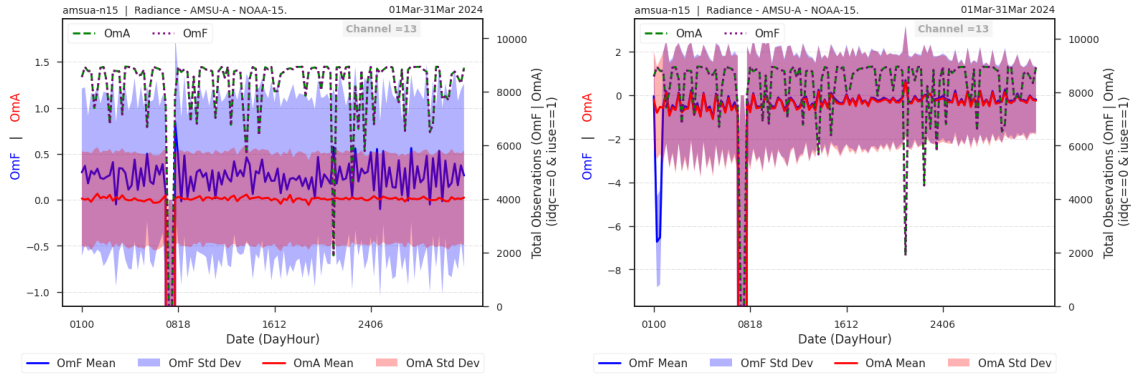


4.3.1 Canal 13

Espera-se que nos experimentos gerados a partir da versão em desenvolvimento do SMNA, os resultados de OmF e OmA sejam melhores. Mais especificamente, pretende-se obter valores pequenos de médias globais destes parâmetros, em magnitude. Além disso, espera-se que uma quantidade maior de dados sejam assimilados nos experimentos mais recentes: **EXP20** e **EXP21**.



(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.



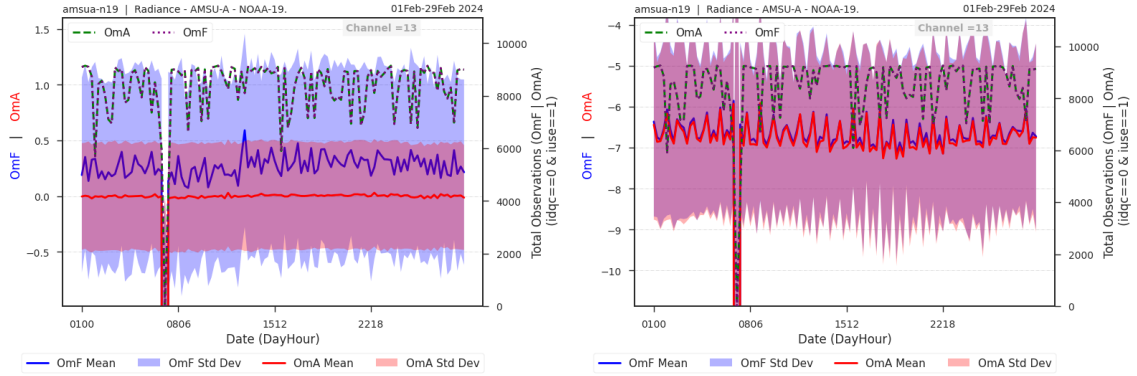
(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

Figura 4.12 - Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-15, canal 13, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.

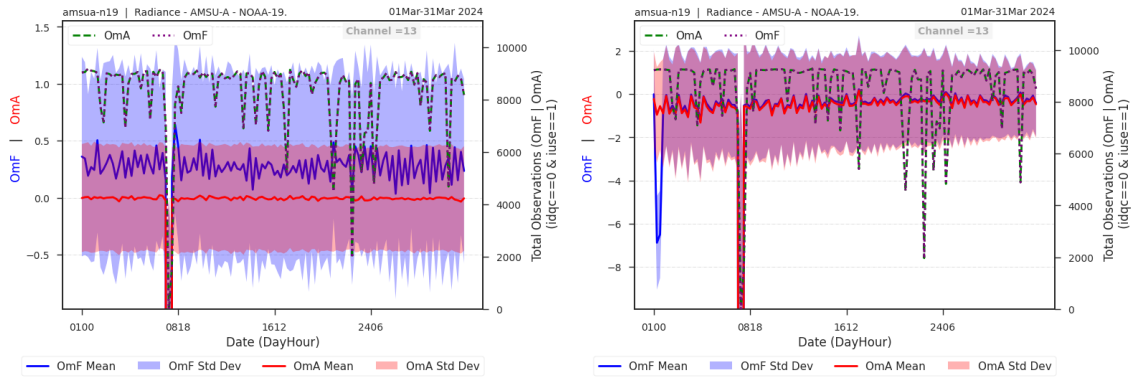
Na Figura 4.12, são plotadas as médias globais e desvio padrão dos dados obtidos pelo sensor AMSU-A no satélite NOAA-15, canal 13. Comparando os experimentos, nota-se que apenas os resultados do **EXP21** apresentam uma tendência de diminuição, em magnitude, das médias de OmF e OmA no decorrer do tempo. O que

indica que ao executar rodadas mais prolongadas do SMNA os resultados obtidos pela versão em desenvolvimento retornaram erros cada vez menores.

Verifica-se que os resultados para o satélite NOAA-19, apresentados na Figura 4.13, tem comportamento similar ao analisado na Figura 4.12 para o satélite NOAA-15: é averiguado um comportamento oscilatório e constante para **EXP19** e **EXP20**, enquanto que, no **EXP21** as médias tendem a zero com o passar do tempo.



(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.



(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

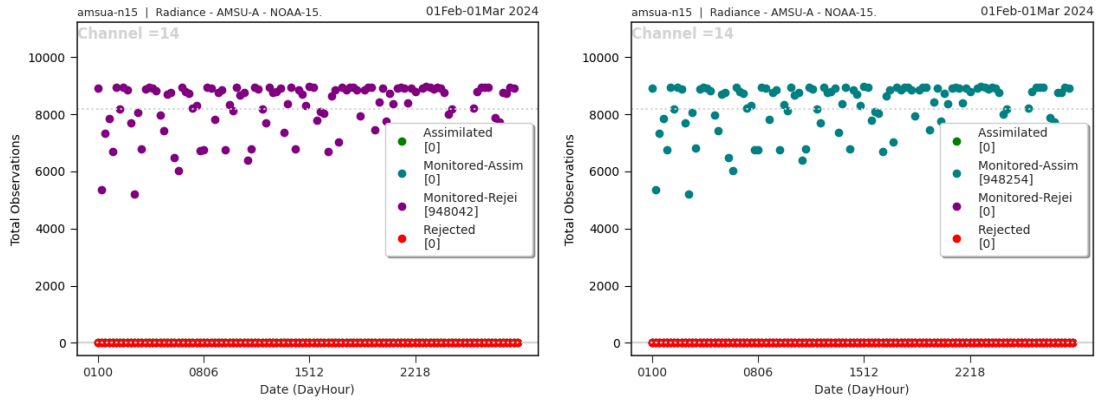
Figura 4.13 - Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-19, canal 13, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.

Uma outra diferença entre os resultados dos experimentos nas Figuras 4.12 e 4.13, pode ser observada. As curvas de média e regiões de desvio dos experimentos 20 e 21 indicam que os valores de OmF e OmA são bem semelhantes/próximos (sobrepostos). No caso do experimento 19, há uma discrepância entre os valores das médias e

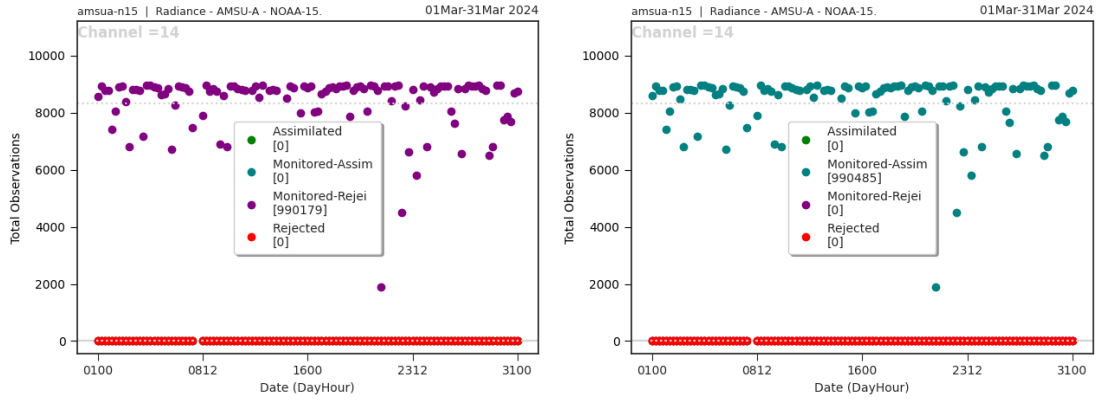
regiões de desvio de OmF e OmA.

4.3.2 Canal 14

O canal 14 apresenta uma particularidade: como pode ser constatado na Tabela 4.2, nos satélites NOAA-18, NOAA-19 e MetOp-b este canal é assimilado, contudo no satélite NOAA-15 o mesmo canal é monitorado. Nas Figuras 4.14, 4.15 e 4.16 são apresentadas as série temporais da quantidade de observações assimiladas, rejeitadas e monitoradas do canal 14 nos satélites NOAA-15-18 e NOAA-19 (o satélite MetOp-b não consta na base de dados do **EXP19** e será analisado em outra seção).



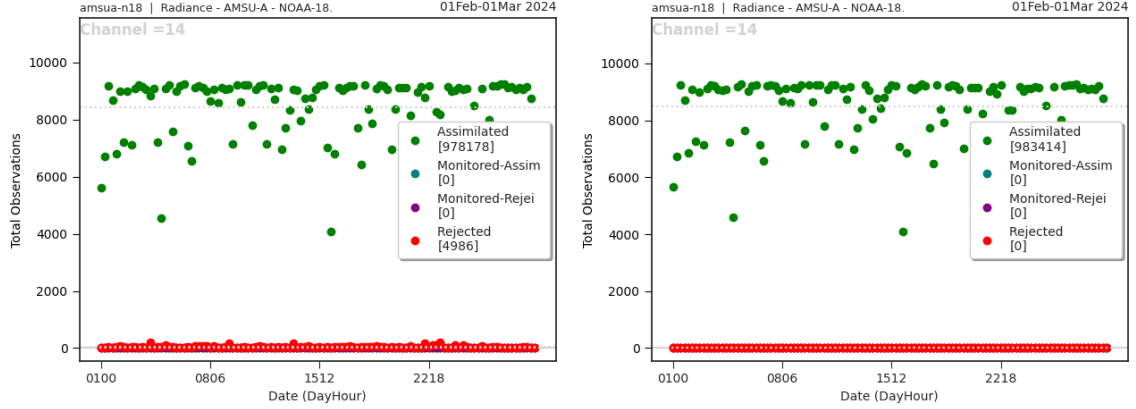
(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.



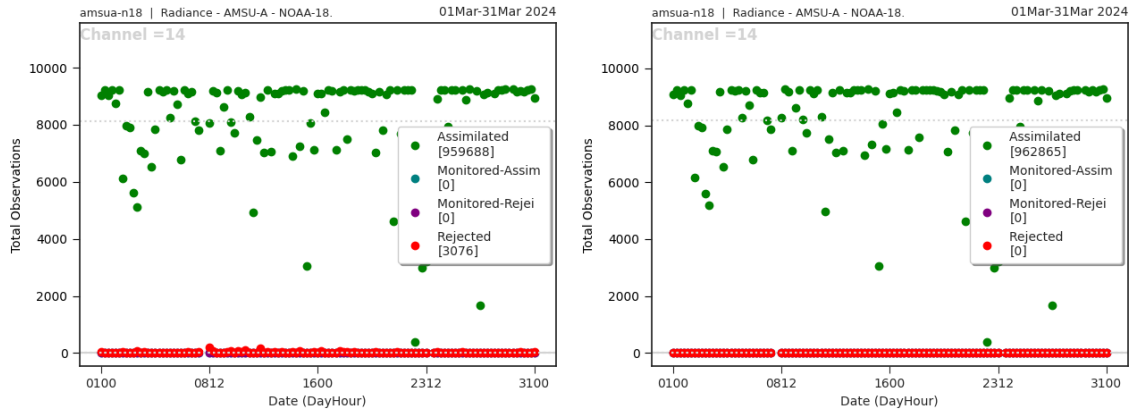
(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

Figura 4.14 - Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) obtidos pelo satélite NOAA-15, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b).

As observações monitoradas, são separadas em dois subgrupos: as observações monitoradas que poderiam ser assimiladas, pois a “tag” do controle de qualidade *idqc* é zero e as observações monitoradas que poderiam ser rejeitadas, pois a “tag” do controle de qualidade *idqc* é um valor diferente de zero.



(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.

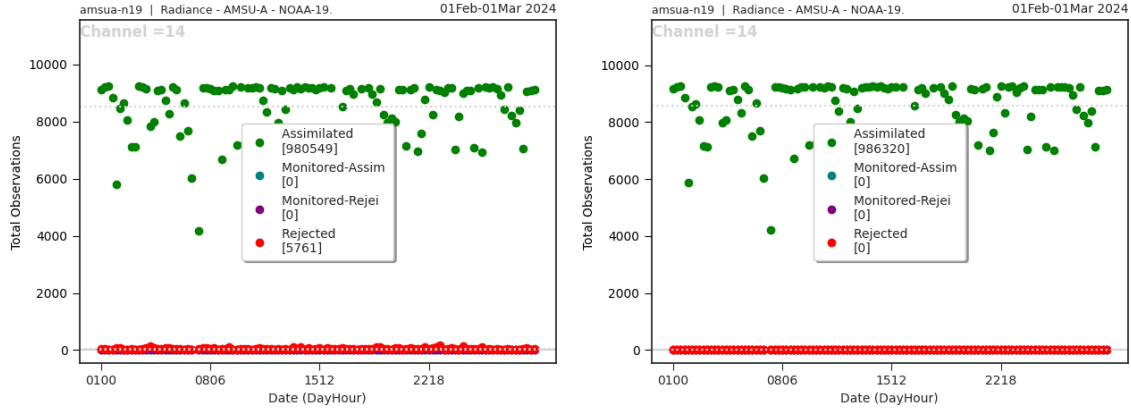


(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

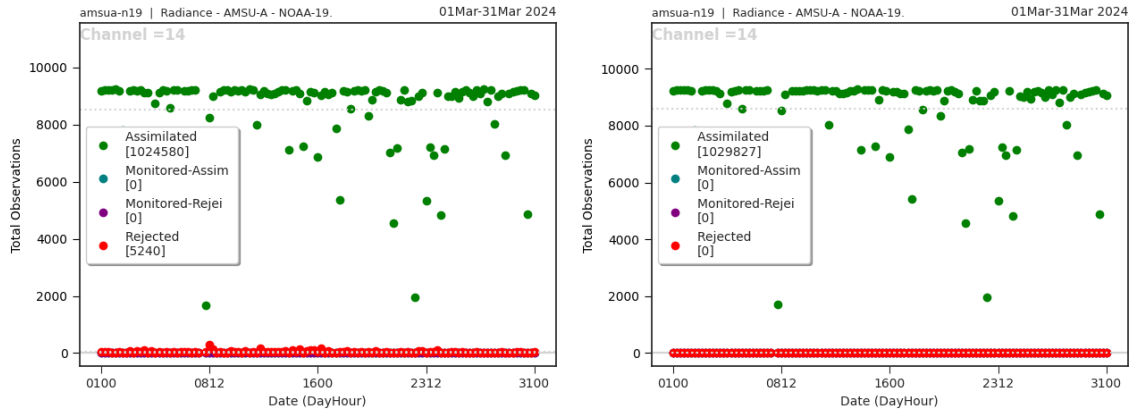
Figura 4.15 - Série temporal do total de observações, assimiladas, rejeitadas e monitoradas (*idqc* = 0 - monitorado assimilado e *idqc* ≠ 0 - monitorado rejeitado) obtidos pelo satélite NOAA-18, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b).

Analisando as Figuras 4.14 - 4.16, é possível constatar que o dados do satélite NOAA-15 (Figura 4.14) são monitorados. Verifica-se que nos experimentos 20 e 21 as observações monitoradas seriam possivelmente assimiladas, uma vez que o índice de controle de qualidade *idqc* é zero, o que indica uma melhora em relação ao experimento 19 que seria completamente rejeitado, pois, *idqc* ≠ 0. Nas Figuras 4.15 e 4.16

observa-se que o canal 14 é assimilado e no experimento 19, há uma quantidade, ainda que pequena, de dados rejeitados. Tal fato não é observado nos experimentos 20 e 21, sinalizando uma melhoria na assimilação de dados, ou seja, mais dados foram assimilados nos experimentos da versão em desenvolvimento do SMNA.



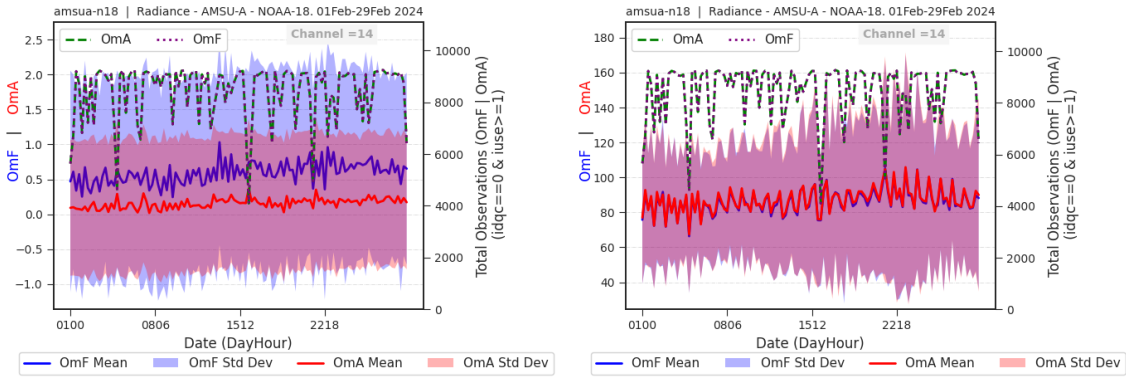
(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.



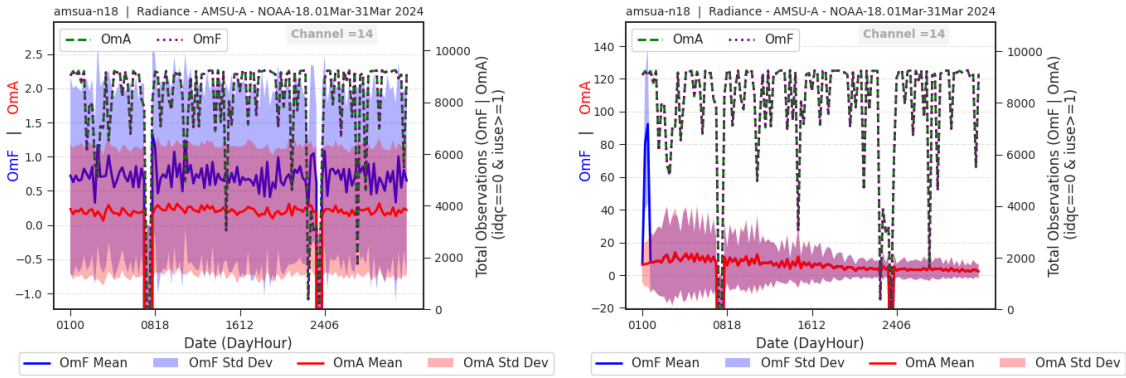
(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

Figura 4.16 - Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) obtidos pelo satélite NOAA-19, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b).

Outro estudo que pode ser realizado utilizando os plotes da função “`time_series_radi`” é uma comparação entre os valores de média e desvio padrão de OmF e OmA durante todo o período dos experimentos **EXP19**, **EXP20** e **EXP21** no canal 14. As Figuras 4.17 e 4.18 apresentam esses resultados.



(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.

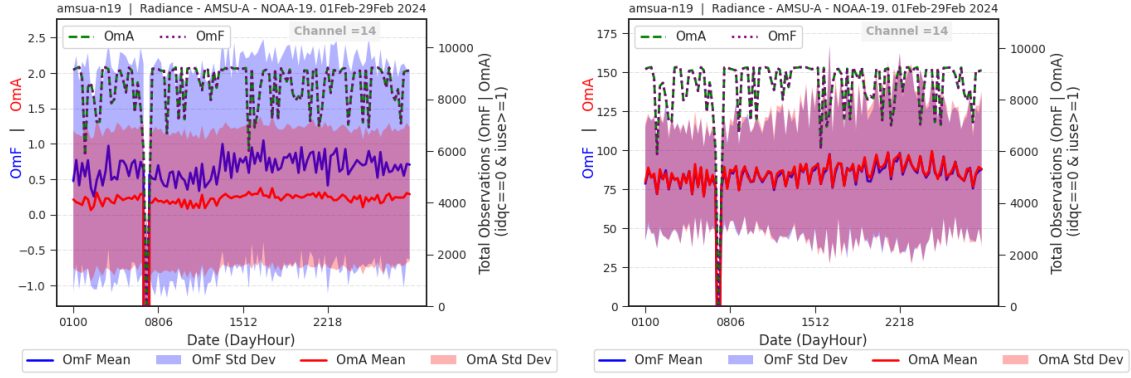


(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

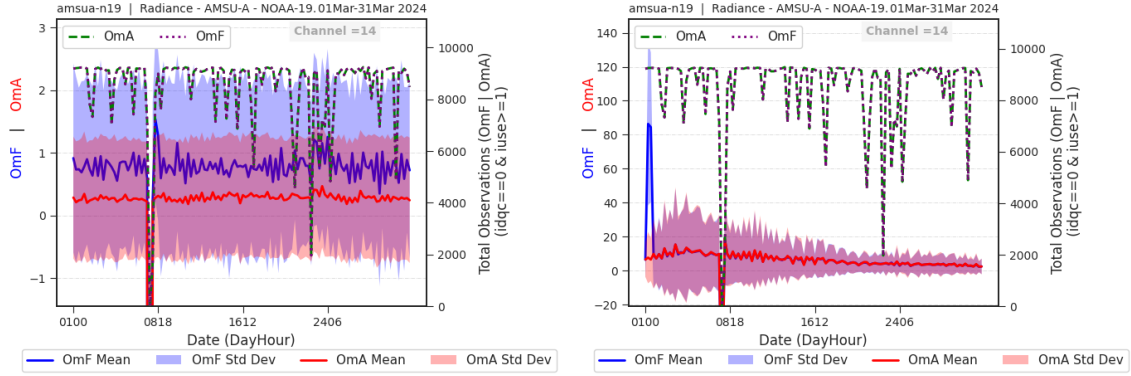
Figura 4.17 - Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-18, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.

Note que o satélite NOAA-15 não é considerado nesta comparação pois o canal 14 é monitorado em todos os experimentos. Como constatado no canal 13, **EXP21**, da Subseção 4.3.1, no canal 14 observa-se que os valores médios de OmF e OmA decrescem com o passar do tempo no experimento 21. Neste canal, mais especificamente no experimentos 20 e 21, há uma sobreposição das curvas de OmF e OmA, bem como sobreposição da região de desvio, em comparação com os resultados obtidos no experimento 19, porém, no experimento 20 o comportamento é oscilante com uma pequena tendência de queda até o dia 08 e na sequência há um aumento nos valores de média chegando ao máximo próximo do dia 22 (ver Figura 4.19), enquanto que no experimento 21 os valores de média decrescem e tendem a zero. Além disso, quantitativamente, é evidente que os valores de média de OmF e OmA no experimento 19 é muito menor comparado aos valores obtidos no experimento

20. Já o experimento 21 apresenta médias inferiores a 20, que se aproximam de zero no final da série temporal, o que indica uma melhora significativa na assimilação de dados de radiância no **EXP21**. Vale lembrar que no experimento 20 não foi realizada a correção de viés o que possivelmente fez com que os resultados da série temporal não apresentasse tendência de queda (diminuição dos valores).

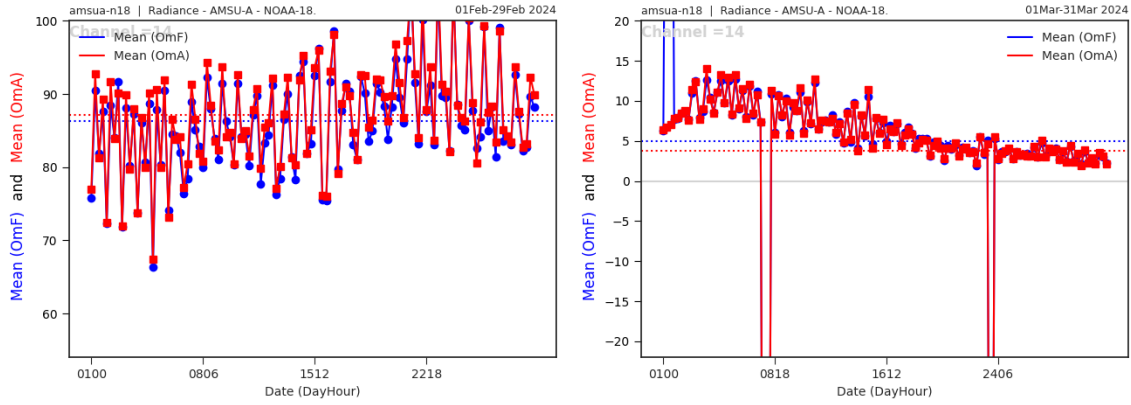


(a) **EXP19** (esquerda) e **EXP20** (direita) - FEV 2024.

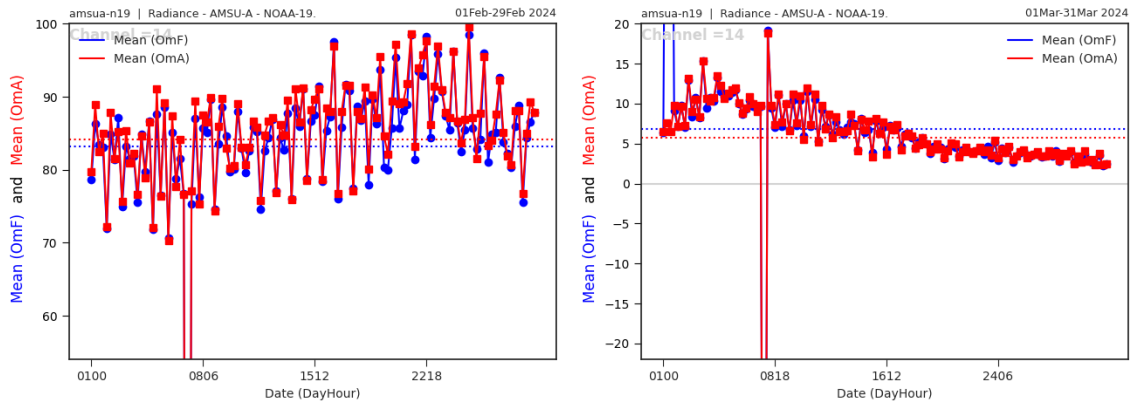


(b) **EXP19** (esquerda) e **EXP21** (direita) - MAR 2024.

Figura 4.18 - Série temporal das médias globais e desvio padrão dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelo satélite NOAA-19, canal 14, nos experimentos 19 e 20 (mês de Fevereiro) (a) e nos experimentos 19 e 21 (mês de Março) (b). As linhas tracejadas fornecem a quantidade de observações assimiladas.



(a) NOAA-18: média OmF e OmA, **EXP20** (esquerda) e **EXP21** (direita).



(b) NOAA-19: média OmF e OmA, **EXP20** (esquerda) e **EXP21** (direita).

Figura 4.19 - Série temporal das médias globais e dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelos satélite NOAA-18 (a) e NOAA-19 (b) do canal 14 no experimento 21.

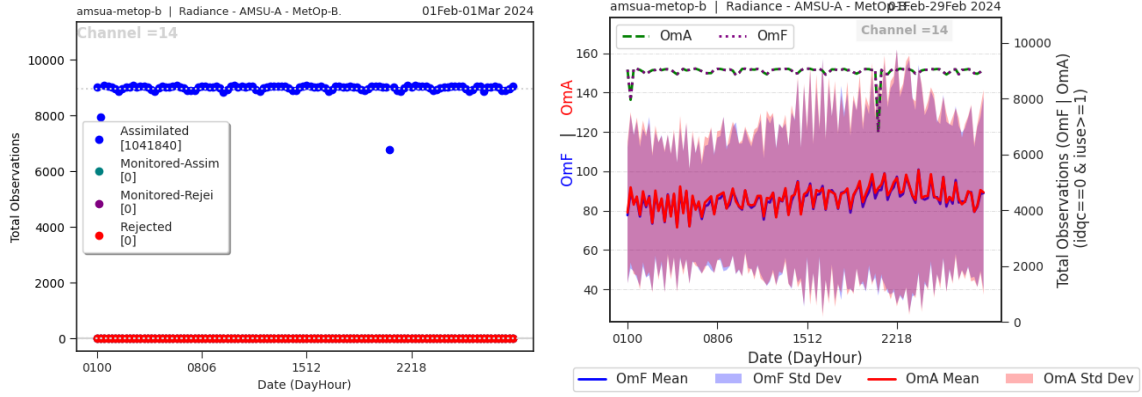
Na base de dados dos experimentos gerados e utilizados neste trabalho para análise diagnóstica da assimilação das observações de radiância, verifica-se que os dados do satélite MetOp-b só foi considerado nos experimentos 20 e 21. Por esse motivo, na próxima seção apresenta-se as séries temporais do diagnóstico da radiância dos dados deste satélite.

4.4 Comparação entre os **EXP20** e **EXP21** do satélite MetOp-b

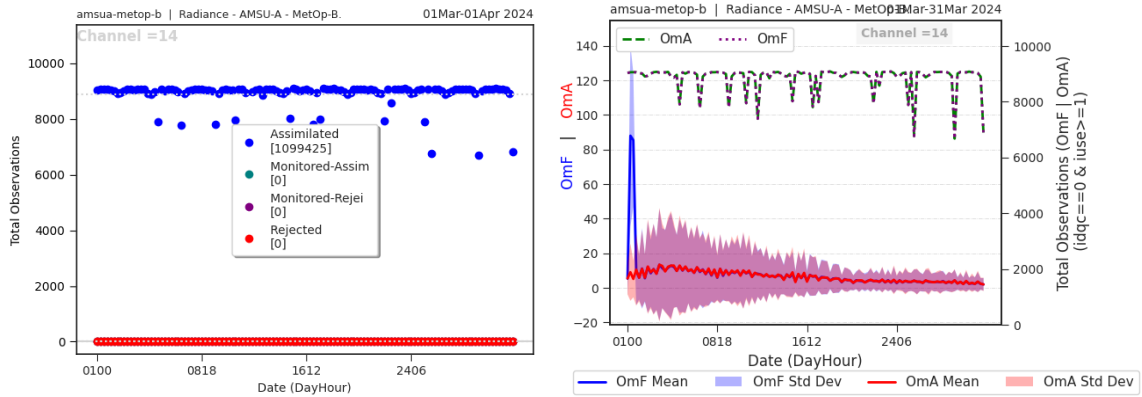
Período: $\begin{cases} \text{EXP20} \text{ --- } > 01/02/2024 \text{ à } 29/02/2024, \\ \text{EXP21} \text{ --- } > 01/03/2024 \text{ à } 31/03/2024. \end{cases}$

Os dados do satélite MetOp-b estão disponíveis apenas nos experimentos **EXP20** e **EXP21**. Embora os períodos dos experimentos sejam diferentes, como constatado

nas subseções anteriores, há uma melhora na assimilação de dados do experimento **EXP21** em relação a assimilação de dados do experimento **EXP19**. Nesta subseção pretende-se averiguar se os dados do satélite MetOp-b tem comportamento similar ao observado nos experimentos 20 e 21 dos satélites NOAA-15-18 e 19. A Figura 4.22 contém o diagrama de *Hovmoller* que mostra as médias e desvio padrão de OmF e OmA de todos os canais do satélite nos experimentos **EXP20** (a) e **EXP21** (b).



(a) **EXP20**: Total de observações (esquerda), média e desvio padrão de OmF e OmA (direita).



(b) **EXP21**: Total de observações (esquerda), média e desvio padrão de OmF e OmA (direita).

Figura 4.20 - Série temporal do total de observações, assimiladas, rejeitadas e monitoradas ($idqc = 0$ - monitorado assimilado e $idqc \neq 0$ - monitorado rejeitado) (esquerda) e série temporal das médias e desvio de OmF e OmA (direita) obtidos pelo satélite MetOp-b, canal 14, no experimento 20 (mês de Fevereiro) (a) e no experimento 21 (mês de Março) (b).

Observa-se que são poucos os canais em que os dados obtidos pelo satélite MetOp-b são assimilados. Para visualizar detalhadamente os valores de média e desvio padrão em um canal específico, apresenta-se nas Figuras 4.20 e 4.21 os resultados

do canal 14, onde verifica-se que todas observações foram assimiladas (ver gráficos à esquerda na Figura 4.20). No experimento **EXP20**, as médias de OmF e OmA oscilam próximo ao valor 80, numa faixa entre 60 e 100 (gráfico à direita da Figura 4.20-(a)). No caso do **EXP21**, novamente, observa-se que os valores das médias decrescem com o passar do tempo e se aproximam de zero, o que indica que o erro entre o valor observado e o valor analisado tende a zero, o que fica ainda mais evidente na Figura 4.21.

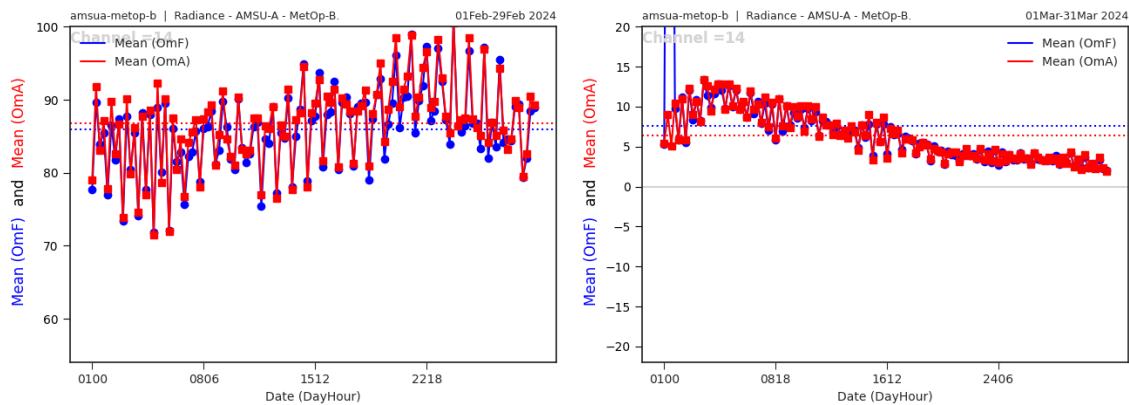
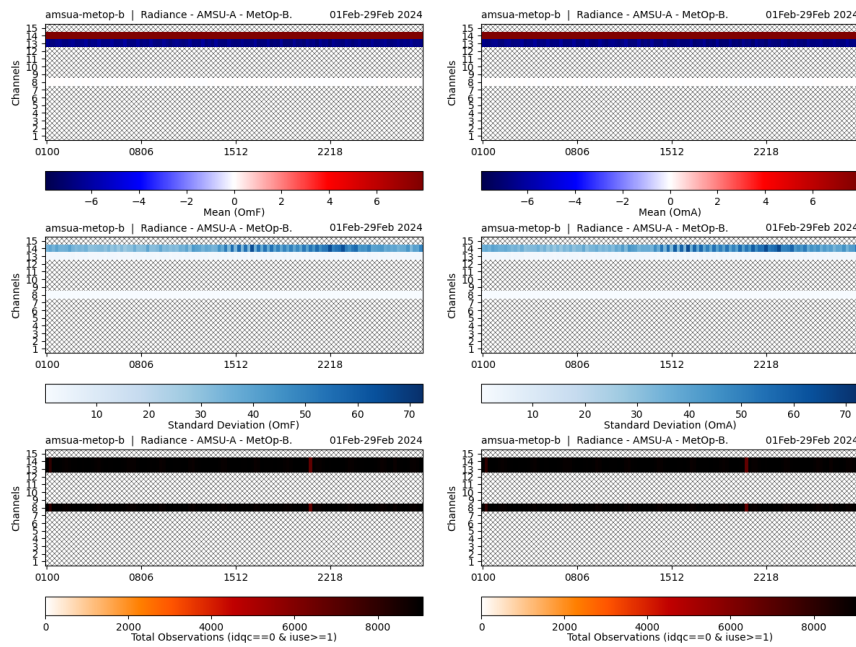
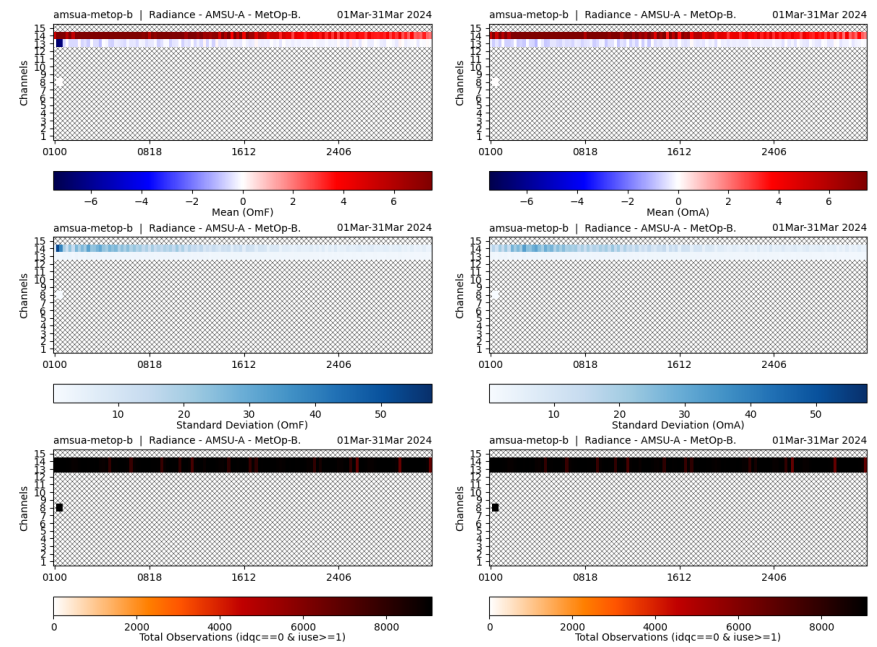


Figura 4.21 - Série temporal das médias globais e dos parâmetros OmF (azul) e OmA (vermelho) obtidos pelos satélite MetOp-b, canal 14, no **EXP20** (esquerda) e **EXP21** (direita).



(a) **EXP20**



(b) **EXP21**

Figura 4.22 - Série temporal das médias globais, desvio padrão e quantidade de dados assimilados dos parâmetros OmF (coluna à esquerda) e OmA (coluna à direita) obtidos pelo satélite MetOp-b nos experimentos 20 (a) e 21 (b).

5 Considerações finais

O desenvolvimento de ferramentas no **readDiag** para a avaliação diagnóstica da assimilação de dados de radiância, foi proposta no Projeto 9.1, intitulado: “Assimilação de dados de radiância no aprimoramento da Previsão Numérica do CPTEC” no âmbito do Programa de Capacitação Institucional (PCI) do CPTEC-INPE. Neste relatório técnico foram apresentadas as primeiras contribuições deste projeto.

Aspectos importantes para o entendimento dos arquivos de diagnóstico da radiância gerados pelo GSI foram descritos no capítulo ???. Em um primeiro momento, entender quais parâmetros e a forma que são descritos nos arquivos de diagnóstico permite criar ferramentas para uma análise contínua dos dados e fornece uma interpretação precisa do impacto dos dados de radiância assimilados na Previsão Numérica do Tempo (PNT). Os parâmetros *iuse* e *idqc* são importantes para identificar os dados que foram assimilados e desta forma calcular estatísticas como média e desvio padrão dos erros OmF (observação - previsão) e OmA (observação - análise). A identificação das observações rejeitadas e monitoradas também depende dos valores destes parâmetros. Como descrito no capítulo ??? ao identificar as observações rejeitadas é possível através do valor do *idqc* compreender o motivo da observação não ser assimilada. Para as observações/canais monitorados (*iuse* = -1), neste trabalho foi sugerida uma subclassificação que analisa, caso a observação/canal venha a ser utilizada (*iuse* = 1) o dado seria assimilado (*monitorado-assimilado*) pois, *idqc* = 0 ou o dado seria rejeitado (*monitorado-rejeitado*) pois, *idqc* ≠ 0.

Uma vez compreendido como utilizar os dados de radiância, na sequência, são descritas as principais modificações e contribuições para o aperfeiçoamento da plataforma **readDiag** no capítulo 3. Para obter séries temporais da densidade de dados assimilados, da média global de OmF e OmA e seus respectivos desvios padrão, foi implementada uma nova função, “**time_series_radi**” (ver [issue #38](#) na página do projeto **readDiag** no GitHub), que fornece diagramas contendo as estatísticas de todos os canais do sensor ou de uma lista de canais pré-estabelecida, além de fornecer gráficos de linha das séries temporais quando o usuário indica um único canal como parâmetro de entrada. Como observado no capítulo de resultados 4, os diagramas gerados pela função “**time_series_radi**” permite identificar no intervalo de tempo estabelecido quais canais contém observações assimiladas pelo modelo SMNA, e se essas observações assimiladas impactaram de forma positiva o erro OmA, ou seja, se os valores absolutos de OmA tendem a zero. Identificado os canais e observações assimiladas, bem como sua contribuição na PNT, é importante averiguar se

as demais observações foram rejeitadas ou monitoradas. Para tal finalidade a função “`statcount`” (ver [issue #42](#) na página do projeto **readDiag** no GitHub) foi modificada afim de gerar uma sequência temporal de mapas com a distribuição das observações classificadas em assimiladas ou rejeitadas e no caso de monitoradas, se essas são, *monitoradas-assimiladas* ou *monitoradas-rejeitadas*. Algumas adequações e correções foram realizadas na plataforma **readDiag**. Na função “`plot`” foi inserida a possibilidade de fixar os valores de máximo e mínimo do colorbar discutido na [issue #39](#) no projeto do **readDiag** do GitHub. Outros detalhes técnicos como correção de bugs (falhas) foram realizadas: correção e atualização da tabela de dicionário ([issue #41](#)); problemas na distribuição de cores do parâmetro `idqc` ([issue #40](#)) e um bug nas funções quando o banco de dados não fornece os dados para uma ou mais datas do intervalo de tempo analisado ([issue #43](#)).

REFERÊNCIAS BIBLIOGRÁFICAS

- HU, M.; GE, G.; ZHOU, C.; STARK, D.; SHAO, H.; NEWMAN, K.; BECK, J.; ZHANG, X. Grid-point statistical interpolation (gsi) users guide version 3.7. Developmental Testbed Center. Available at <http://www.dtcenter.org/com-GSI/users/docs/index.php>, p. 149, 2018. 7
- HU, M.; SHAO, H.; STARK, D.; NEWMAN, K.; ZHOU, C.; ZHANG, X. Grid-point statistical interpolation (gsi) users guide version 3.5. Developmental Testbed Center. Available at <http://www.dtcenter.org/com-GSI/users/docs/index.php>, p. 141, 2016. 6, 7
- OLIVEIRA, C. M. L. d.; SAPUCCI, L. F.; BASTARZ, C. F.; MATTOS, J. G. Z. d.; KUBOTA, P. Y.; BAÑOS, I. H.; EICHHOLZ, C.; GALANTE, R.; SILVEIRA, B. B. Assimilação de dados de radiação do sensor amsu-a no gsi: Visão geral sobre sua importância para a qualidade das condições iniciais na pnt do cptec. assimilação de dados de radiação do sensor amsu-a no gsi/bam/cptec: Descrição e importância para a pnt. **Revista Brasileira de Meteorologia**, SciELO Brasil, v. 37, n. 3, p. 347–364, 2022. 4
- SAPUCCI, L. F.; BASTARZ, C. F.; MATTOS, J. G. Z. d.; BAÑOS, I. H.; EICHHOLZ, C.; SANTOS, W. H. d.; NEGRI, R. G.; FRASSONI, A. **Avaliação diagnóstica da evolução do SMNA (Versão 2.3.1)**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2024. 163 p. Disponível em: <http://urlib.net/ibi/8JMKD3MGP3W34R/3TAMPU2>. Acesso em: 20 mar. 2024. 18

Notebook tutorial: readDiag radiância

readDiag_Radiancia_tutorial_completo-pt_br

January 16, 2025

1 readDiag - Radiância

O pacote `readDiag` foi concebido para ser uma ferramenta que possibilita o fácil acesso aos arquivos diagnósticos gerados pelo [Gridpoint Statistical Interpolation \(GSI\) system](#). O pacote é composto principalmente por 2 classes, a primeira para ler os dados (`read_diag`) e outra para fazer diferentes tipos de figuras com dados selecionados (`plot_diag`). O detalhamento das ferramentas disponíveis nas classes `read_diag` e `plot_diag` encontra-se no notebook `readDiag_tutorial_completo-pt_br`. Neste notebook apresenta-se as ferramentas do `readDiag` construídas e modificadas (até o momento) para acesso aos arquivos de diagnóstico da **radiância** gerados pelo GSI.

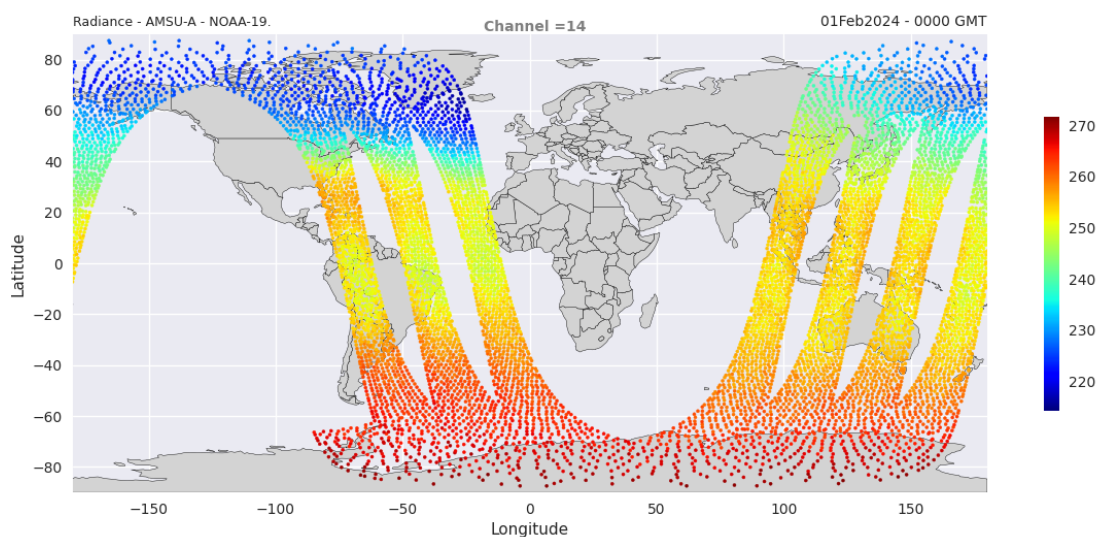
1.1 A Classe `read_diag`

A classe `read_diag` não foi alterada. Abaixo as ferramentas desta classe são utilizadas como exemplo para visualização do diagnóstico de dados de **radiância**.

1.2 A Classe `plot_diag`

Na classe `plot_diag` as funções abaixo foram construídas ou modificadas para leitura e análise dos dados de **radiância**:

1. `plot(self, varName, varType, param, minVal=None, maxVal=None, mask=None, area=None, **kwargs)`: a função `plot` gera uma figura para a variável `varName` (ex: `amsua`), `varType` (ex: `n19` (satélite)) e `param`, que pode ser várias opções como: `param="obs"` para o valor da observação, `param="omf"` para observação menos background ou `param="oma"` para observação menos análise. Foram inseridas as variáveis opcionais `minVal` e `maxVal` para fixar o range de cores do colormap. Também é possível mascarar os dados com as variáveis `iuse` e `idqc`, que indicam se o dado de **radiância** foi (`iuse>=1 & idqc=0`) assimilado ou rejeitado (`iuse>=1 & idqc!=0`), ou ainda, se o dado é monitorado (`iuse=-1`). Utilizando o `mask` ainda é possível especificar o canal `nchan` (n^o do canal). Abaixo há um exemplo executado para `varName="amsua"`, `varType="n19"`, `param="obs"` e `mask="(nchan==14) & (iuse >= 1 & idqc == 0)"`;



2. `time_series_radi(self, varName=None, varType=None, mask=None, dateIni=None, dateFin=None, nHour="06", vminOMA=None, vmaxOMA=None, vminSTD=0.0, vmaxSTD=14.0, channel=None, Clean=None)`: a função `time_series_radi` gera figuras que mostram a variação temporal das médias e desvios padrões dos parâmetros OmF e OmA dos dados de **radiância**. Essa função pode ser utilizada considerando-se três situações: quando `channel` é o nº de um canal (4 Figuras são geradas utilizando apenas as observações do canal especificado), quando `channel` é uma lista de canais (6 diagramas de *Hovmoller* são criados considerando apenas os canais listados) e quando `channel` é `None` (são gerados 6 diagramas de *Hovmoller* considerando todos os canais disponíveis (1 à 15)).

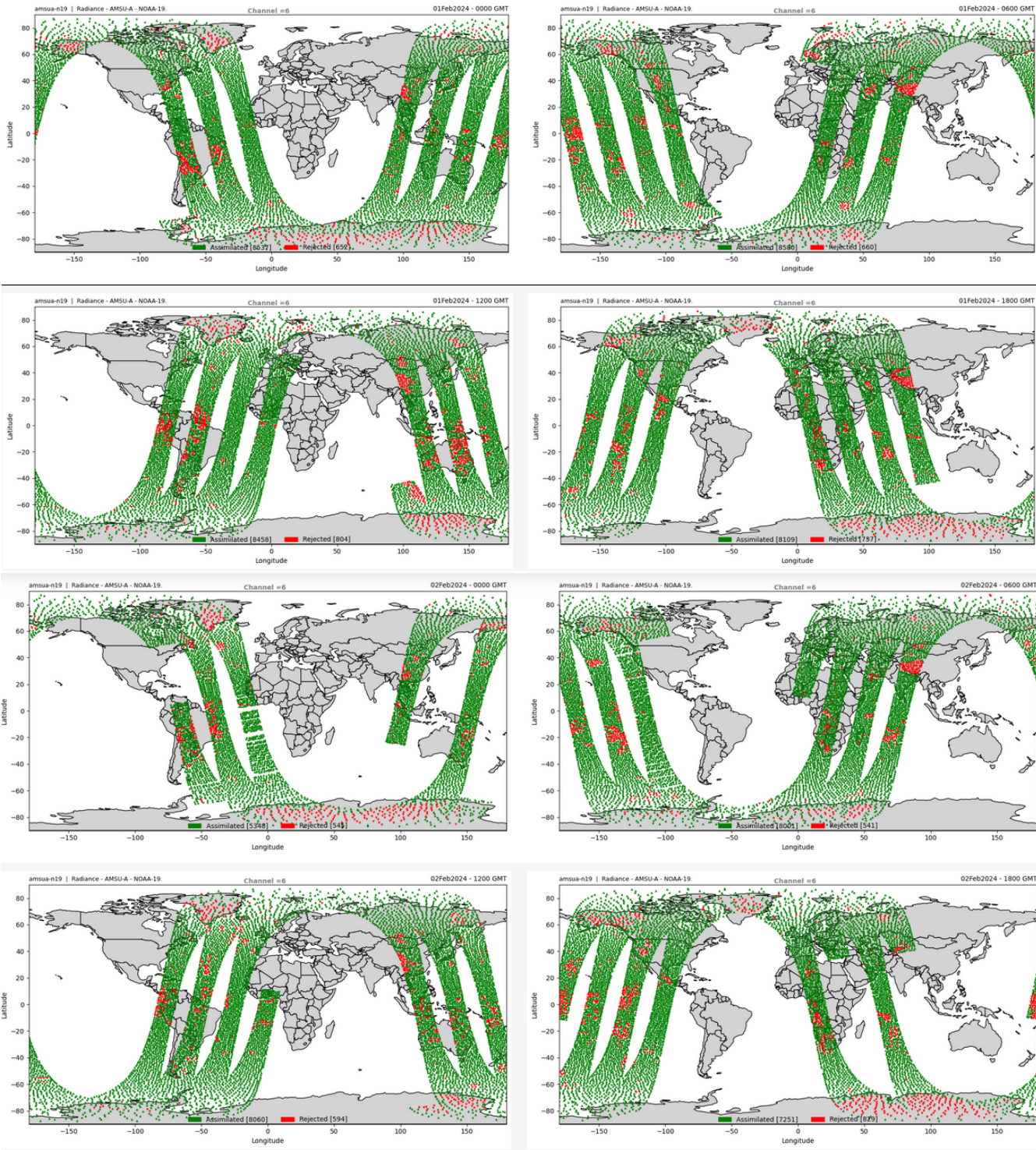
A seguir são apresentados os parâmetros de entrada na função.

Parâmetro	Exemplo	Descrição
<code>self</code>	<code>['/home/user/ diag_amsua_n19_01.2024020100']</code>	Lista com todos os caminhos completos (<code>caminho/nome_do_arquivo</code>) de cada tempo da série temporal.
<code>varName</code>	<code>amsua</code>	Nome da variável
<code>varType</code>	<code>n19</code>	Tipo da variável
<code>dateIni</code>	<code>2024020100</code>	Data inicial da série temporal
<code>dateFin</code>	<code>2024020218</code>	Data final da série temporal
<code>nHour</code>	<code>6</code>	Intervalo de tempo em horas entre cada arquivo na lista <code>self</code>
<code>vminOMA</code>	<code>-2.0</code>	Valor mínimo da escala y (ordenada) para OmF e OmA
<code>vmaxOMA</code>	<code>2.0</code>	Valor máximo da escala y (ordenada) para OmF e OmA
<code>vminSTD</code>	<code>0.0</code>	Valor mínimo da escala y (ordenada) para o desvio-padrão
<code>vmaxSTD</code>	<code>14.0</code>	Valor máximo da escala y (ordenada) para o desvio-padrão
<code>channel</code>	<code>14</code>	Valor(es) do canal a ser feita a série temporal, opções: valor numérico referente ao canal, ex.: 14 para plotar um canal; [6, 9, 10, 13, 14] para plotar uma lista de canais específica; <code>None</code> para plotar todos os canais.
<code>Clean</code>	<code>True</code> ou <code>False</code>	Se <code>True</code> , após gerar e salvar a figura, a janela da figura é reiniciada (<code>plt.clf()</code>) ou fechada (<code>plt.close()</code>), se <code>False</code> , este procedimento é eliminado e a figura continua disponível para visualização com <code>plt.show()</code> .

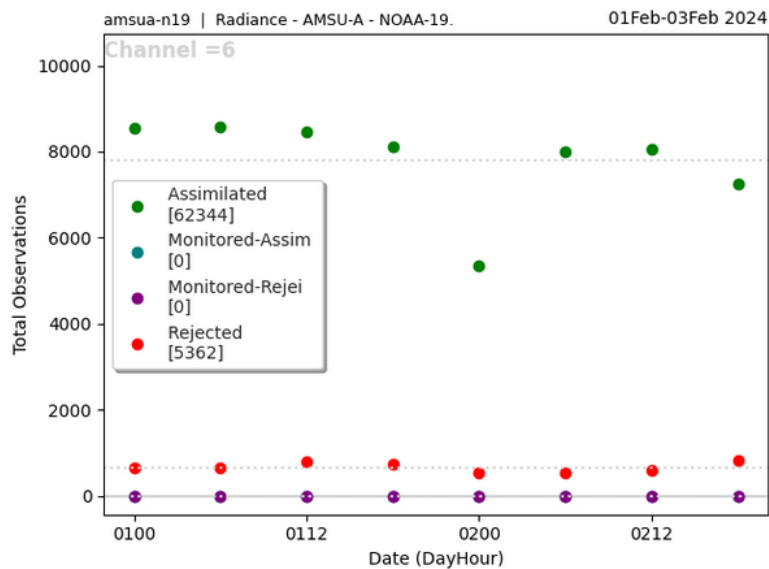
3. `statcount(self, varName=None, varType=None, noiqc=False, dateIni=None, dateFin=None, nHour="06", channel=None, figTS=False, figMap=False, **kwargs)`: a função `statcount` gera dois tipos de figuras. Se `figTS` é `True` uma figura com a série temporal da quantidade de dados assimilados, rejeitados e monitorados é gerada. Se `figMap` é `True` uma sequência temporal de figuras indicando no mapa a geolocalização das observações assimiladas, rejeitadas e monitoradas são produzidas. Nesta função, quando os dados são de **radiância**, a variável `channel` recebe o número específico de um canal e quando se trata de dados convencionais a variável `channel` deve ser `None`. Quando `channel = nº canal` os dados monitorados (`iuse=-1`) são subclassificados em dois grupos: observações monitoradas que seriam assimiladas (monitorado-assimilado) pois, `idqc=0` e observações monitoradas que seriam rejeitadas (monitorado-rejeitado) pois, `idqc!=0`. Logo, para os dados de **radiância**, se `figMap=True` podem ser geradas duas figuras, uma com a geolocalização das observações assimiladas e rejeitadas e outra com a geolocalização das observações monitoradas-assimiladas e monitoradas-rejeitadas. A variável `noiqc` é um parâmetro da lista de nomes do GSI sendo utilizada na classificação dos dados convencionais e pode receber os

valores True ou False (não interfere na classificação da **radiância**). Abaixo há um exemplo executado para `varName="amsua"`, `varType="n19"`, `noiqc=False`, `dateIni="2024020100"`, `dateFin="2024020218"`, `nHour="06"`, `channel=6`, `figTS=True`, `figMap=True`.

`figMap=True`



`figTS=True`



No decorrer deste notebook são mostrados exemplos com recortes de código para exemplificar o uso das funções acima elencadas.

1.3 Utilização da classe `read_diag`

1.3.1 Bibliotecas necessárias

Para iniciar a utilização do `readDiag`, carregue primeiro as bibliotecas necessárias para a sua utilização:

- `gsidiag`: é a biblioteca que contém as classes `read_diag` e `plot_diag`;
- `pandas`: é a biblioteca que fornece as estruturas de dados tabulados utilizadas pelo `readDiag`;
- `matplotlib`: é a biblioteca a partir da qual são confeccionadas as figuras;
- `datetime`: é a biblioteca utilizada para manipular datas;
- `datasources`: é a biblioteca utilizada para analisar e armazenar informações de um arquivo contendo dados de observações;
- `os`: é a biblioteca que fornece diversas interfaces do sistema operacional.

A instrução `%matplotlib inline` é um comando mágico do Jupyter e apenas ajusta o ambiente para que não seja necessário utilizar o comando `plt.show()` sempre que figuras forem mostradas dentro do notebook. Se você estiver utilizando o `readDiag` dentro de um script Python, esta diretiva pode ser suprimida e o comando `plt.show()` deve ser utilizado, a depender da situação.

```
[ ]: # Descomente a linha a seguir para utilizar este notebook no Google Colab
      #!pip install readDiag
```

```
[1]: import gsidiag as gd

import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from gsidiag.datasources import getVarInfo

import os

%matplotlib inline
```

1.3.2 Variáveis principais

Na tabela a seguir, estão relacionadas os valores padrão das variáveis utilizadas ao longo deste tutorial. Observe que as variáveis possuem diferentes tipos e que algumas podem ser declaradas como None.

Variáveis	Valores	Tipo
DIRdiag	= /pesq/share/das/dist/joao.gerd/EXP18/GSI/dataout	string
varName	= amsua	string
varType	= n19	string
dateIni	= 2024021000	string
dateFin	= 2024021018	string
nHour	= 6	string
vminOMA	= -2.0	float
vmaxOMA	= 2.0	float
vminSTD	= 0.0	float
vmaxSTD	= 14.0	float
channel	= 6	integer ou None

Dessa forma, na célula a seguir são ajustadas as variáveis seguindo os valores da tabela acima. Além disso, são ajustados também os valores das datas para compor o caminho e os nomes dos arquivos de diagnóstico do GSI:

```
[2]: #Endereço onde os dados devem estar disponiveis para os testes
#Para uso na Egeon
#DIRdiag = "/pesq/share/das/dist/joao.gerd/EXP18/GSI/dataout"

# Para uso na Itapemirim (Colorado, Ilopolis e outras máquinas virtuais)
#DIRdiag = "/share/das/dist/joao.gerd/EXP18/GSI/dataout"

# Para uso em máquina local (neste caso, o diretório remoto deverá ser montado localmente com o
↪ comando sshfs)
# Exemplo:
# $ sshfs username@egeon.cptec.inpe.br:/pesq/share/das/dist/joao.gerd/EXP18/GSI/dataout /extra2/
↪ EGEON_EXP18_GSI_dataout
# ou
# $ sshfs username@itapemirim.cptec.inpe.br:/share/das/dist/joao.gerd/EXP18/GSI/dataout /extra2/
↪ EGEON_EXP18_GSI_dataout
# DIRdiag = '/extra2/EGEON_EXP18_GSI_dataout'

# Para maquinas fora do cptec é preciso fazer o download dos dados fora python, para isso pode ser
↪ feito os seguintes comandos

# cd ~/readDiag/data
# EXP19: Para executar os dois dias são necessários pelo menos 2GB de memória RAM
# wget -c -r https://ftp1.cptec.inpe.br/pesquisa/das/testecase/SMNAexp19_amsua.tar.gz
# tar -xvzf ftp1.cptec.inpe.br/pesquisa/das/testecase/SMNAexp19_amsua.tar.gz
# rm -rf ftp1.cptec.inpe.br
# !wget -P ../data -c -r https://dataserver.cptec.inpe.br/dataserver_dimmt/das/testecase/
↪ SMNAexp19_amsua.tar.gz
# !wget -P ../data -c -r https://dataserver.cptec.inpe.br/dataserver_dimmt/das/testecase/
↪ SMNAexp19_conv_0201.tar.gz
# !wget -P ../data -c -r https://dataserver.cptec.inpe.br/dataserver_dimmt/das/testecase/
↪ SMNAexp19_conv_0202.tar.gz

# !tar -xvzf ../data/dataserver.cptec.inpe.br/dataserver_dimmt/das/testecase/SMNAexp19_amsua.tar.
↪ gz -C ../data/dataserver.cptec.inpe.br/dataserver_dimmt/das/testecase/
```

```
# !tar -zxvf ../data/dataserver.cptec.inpe.br/dataserver_dimnt/das/testecase/SMNAexp19_conv_0201.
→tar.gz -C ../data/dataserver.cptec.inpe.br/dataserver_dimnt/das/testecase/
# !tar -zxvf ../data/dataserver.cptec.inpe.br/dataserver_dimnt/das/testecase/SMNAexp19_conv_0202.
→tar.gz -C ../data/dataserver.cptec.inpe.br/dataserver_dimnt/das/testecase/

# To obtain the ${USER} from the operating system, the os.environ.get('USER') function from the os_
→library is used.
# DIRdiag = os.path.join(os.getcwd(), '../data/dataserver.cptec.inpe.br/dataserver_dimnt/das/
→testecase/SMNAexp19/GSI/dataout')
DIRdiag = os.path.join(os.getcwd(), '../../readDiag/data/SMNAexp19/GSI/dataout')

varName = 'amsua'
varType = 'n19'
dateIni = '2024020100'
dateFin = '2024020218'
nHour = '6'
vminOMA = -2.0
vmaxOMA = 2.0
vminSTD = 0.0
vmaxSTD = 14.0
channel = 6

datei = datetime.strptime(str(dateIni), '%Y%m%d%H')
datef = datetime.strptime(str(dateFin), '%Y%m%d%H')
dates = [dates.strftime('%Y%m%d%H') for dates in pd.date_range(datei, datef, freq='6H').tolist()]

print(dates)
```

```
['2024020100', '2024020106', '2024020112', '2024020118', '2024020200',
'2024020206', '2024020212', '2024020218']
```

Gerando as variáveis `path` e `pathsc` onde estarão os caminhos completos (incluindo nome do arquivo) dos arquivos diagnósticos do primeiro (OmF) e último (OmA) outer loop dos dados de **radiância**:

```
[3]: # Adaptado para criar os caminhos para arquivos de radiância!
paths, pathsc = [], []

OuterL = '01'
[paths.append(DIRdiag + '/' + dt + '/diag_' + varName + '_' + varType + '_' + OuterL + '.' + dt) for_
→dt in dates]

OuterLc = '03'
[pathsc.append(DIRdiag + '/' + dt + '/diag_' + varName + '_' + varType + '_' + OuterLc + '.' + dt)
→for dt in dates]

print(paths)
print('')
print(pathsc)
```

```
['/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GS
I/dataout/2024020100/diag_amsua_n19_01.2024020100', '/home/radiancia/DiagRadi-V2
_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020106/diag_ams
ua_n19_01.2024020106', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../read
Diag/data/SMNAexp19/GSI/dataout/2024020112/diag_amsua_n19_01.2024020112', '/home
/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/datao
ut/2024020118/diag_amsua_n19_01.2024020118', '/home/radiancia/DiagRadi-V2_issue4
5/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020200/diag_amsua_n19_
```

```
01.2024020200', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020206/diag_amsua_n19_01.2024020206', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020212/diag_amsua_n19_01.2024020212', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020218/diag_amsua_n19_01.2024020218']
```

```
['/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020100/diag_amsua_n19_03.2024020100', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020106/diag_amsua_n19_03.2024020106', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020112/diag_amsua_n19_03.2024020112', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020118/diag_amsua_n19_03.2024020118', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020200/diag_amsua_n19_03.2024020200', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020206/diag_amsua_n19_03.2024020206', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020212/diag_amsua_n19_03.2024020212', '/home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020218/diag_amsua_n19_03.2024020218']
```

1.3.3 Leitura dos arquivos de diagnósticos

Lendo os arquivos diagnósticos de **radiância** com a função `read_diag()` do pacote `readDiag`. No trecho de código a seguir, observe que a função `read_diag()` é utilizada dentro de um loop que varia sobre todos os arquivos das listas `paths` e `pathsc` definidas no passo anterior. No final do loop, é gerada a lista `gdf_list` que conterá todos os arquivos lidos pelo `readDiag`:

```
[4]: read = True

if read:
    gdf_list = []
    print('')

    print('Aguarde, o tempo total estimado para a leitura dos arquivos é de ' +
          str(int((float(len(paths))*20)/60)) + ' minutos e ' +
          str(int((float(len(paths))*20)%60)) + ' segundos.')

    print('')

    for path, pathc in zip(paths, pathsc):
        print('Reading ' + path)

        gdf = gd.read_diag(path, pathc)

        gdf_list.append(gdf)

    print('Pronto!')
```

Aguarde, o tempo total estimado para a leitura dos arquivos é de 2 minutos e 40 segundos.

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020100/diag_amsua_n19_01.2024020100
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020106/diag_amsua_n19_01.2024020106
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020112/diag_amsua_n19_01.2024020112
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020118/diag_amsua_n19_01.2024020118
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020200/diag_amsua_n19_01.2024020200
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020206/diag_amsua_n19_01.2024020206
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020212/diag_amsua_n19_01.2024020212
```

```
>>> GSI DIAG <<<
```

```
Reading /home/radiancia/DiagRadi-V2_issue45/notebooks/../../readDiag/data/SMNAexp19/GSI/dataout/2024020218/diag_amsua_n19_01.2024020218
```

```
>>> GSI DIAG <<<
```

Pronto!

A variável `gdf_list` é uma lista de dataframes contendo os dados de cada arquivo diagnóstico. Para trabalhar com um único tempo basta informar a lista com o índice fixo, por exemplo: `gdf_list[0]`:

```
[5]: gdf_list
```

```
[5]: [<gsidiag.__main__.read_diag at 0x7f39ee0b1e80>,  
      <gsidiag.__main__.read_diag at 0x7f3a1b7e9a60>,  
      <gsidiag.__main__.read_diag at 0x7f3a1b7fd640>,  
      <gsidiag.__main__.read_diag at 0x7f39ee0a4d00>,  
      <gsidiag.__main__.read_diag at 0x7f3a1b7fd610>,  
      <gsidiag.__main__.read_diag at 0x7f39ee0a41f0>,  
      <gsidiag.__main__.read_diag at 0x7f39ee09a8b0>,  
      <gsidiag.__main__.read_diag at 0x7f39f0c31f10>]
```

Fazendo `tidx = 0`, obtemos o primeiro objeto da lista `gdf_list`:

```
[6]: tidx = 0  
      gdf_list[tidx]
```

```
[6]: <gsidiag.__main__.read_diag at 0x7f39ee0b1e80>
```

1.3.4 Obtendo informações dos arquivos

Utilize a função `pfileinfo()` para obter informação como o tipo de sensor e seu respectivo satélite (`n19`) que estão contidos dentro do arquivo:

```
[7]: gdf_list[tidx].pfileinfo()
```

Variable Name : amsua
 kx => n19

Além do método `pfileinfo()`, outros métodos e funções também podem ser utilizados para acessar as informações sobre os arquivos abertos. Para obter uma lista dos métodos e funções disponíveis, digite `gdf_list[tidx].` e pressione a tecla `<TAB>` duas vezes no teclado:

```
>>> gdf_list[tidx].
gdf_list[tidx].close(      gdf_list[tidx].obsInfo      gdf_list[tidx].pfileinfo(  gdf_list[tidx].zlevs
gdf_list[tidx].tocsv(      gdf_list[tidx].overview(   gdf_list[tidx].varNames
gdf_list[tidx].obs
```

Os métodos e funções embutidas possuem documentação, as quais podem ser acessadas da seguinte forma:

```
print(object.function_name.__doc__)
```

ou

```
help(object.function_name)
```

Por exemplo:

```
[8]: print(gdf_list[tidx].pfileinfo.__doc__)
```

Prints a fancy list of the existing variables and types.

Usage: `pfileinfo()`

ou ainda:

```
[9]: help(gdf_list[tidx].pfileinfo)
```

Help on method `pfileinfo` in module `gsidiag.__main__`:

`pfileinfo()` method of `gsidiag.__main__.read_diag` instance
Prints a fancy list of the existing variables and types.

Usage: `pfileinfo()`

Para obter um dicionário com todas as informações sobre as variáveis e tipos contidos no arquivo, utilize o método `obsInfo`:

```
[10]: gdf_list[tidx].obsInfo
```

```
[10]: {'amsua':
```

			lat	lon	elev	nchan	time
iuse	idqc	\					
	SatId	points					
n19	0	-87.446198	6.644900	2647.032227	1.0	2.885000	1.0 8.0
	1	-87.446198	6.644900	2647.032227	2.0	2.885000	1.0 8.0
	2	-87.446198	6.644900	2647.032227	3.0	2.885000	1.0 8.0

```

3      -87.446198      6.644900  2647.032227      4.0  2.885000      1.0   8.0
4      -87.446198      6.644900  2647.032227      5.0  2.885000      1.0   8.0
...
137830 56.754601  336.250885      0.055578  11.0 -1.522778      1.0   0.0
137831 56.754601  336.250885      0.055578  12.0 -1.522778      1.0   0.0
137832 56.754601  336.250885      0.055578  13.0 -1.522778      1.0   0.0
137833 56.754601  336.250885      0.055578  14.0 -1.522778      4.0   0.0
137834 56.754601  336.250885      0.055578  15.0 -1.522778      1.0  51.0

```

```

              inverr      oer      obs      omf      omf_nobc      emiss  \
SatId points
n19  0      0.000000      NaN  187.330002      2.591137      -7.817700  0.813564
      1      0.000000      NaN  189.190002      4.114823      -3.598879  0.801515
      2      0.000000      NaN  211.460007      -1.666820      2.795111  0.791668
      3      0.000000      NaN  225.820007      0.495127      -0.318058  0.791438
      4      0.000000      NaN  229.089996      0.241357      -1.846274  0.790830
...
137830 2.499857  0.400023  219.949997      0.632804      -0.609745  0.856901
137831 1.817697  0.550150  219.039993      0.488474      -0.474023  0.856901
137832 1.248405  0.801043  217.449997      -1.116789      -1.843264  0.856901
137833 0.248456  4.025372  219.979996      -0.762485      -0.762485  0.856901
137834 0.000000      NaN  229.039993 -23.291994 -41.154434  0.950000

```

```

              oma      oma_nobc      imp      dfs  \
SatId points
n19  0      2.803686      -7.640147      NaN      NaN
      1      4.388880      -3.414203      NaN      NaN
      2     -1.364057      3.049402      NaN      NaN
      3      0.594996      -0.230114      NaN      NaN
      4      0.297319      -1.806047      NaN      NaN
...
137830 0.451392      -0.792813      -0.491687 -0.286979
137831 0.409210      -0.554484      -0.129336 -0.070378
137832 -0.789088      -1.543222      -0.779681 -0.456870
137833 0.452604      0.452604      -0.093540 -0.230162
137834 -23.673958 -41.430882      NaN      NaN

```

```

              geometry
SatId points
n19  0      POINT (6.64490 -87.44620)
      1      POINT (6.64490 -87.44620)
      2      POINT (6.64490 -87.44620)
      3      POINT (6.64490 -87.44620)
      4      POINT (6.64490 -87.44620)
...
137830 POINT (-23.74915 56.75460)
137831 POINT (-23.74915 56.75460)
137832 POINT (-23.74915 56.75460)
137833 POINT (-23.74915 56.75460)
137834 POINT (-23.74915 56.75460)

```

[137835 rows x 18 columns]}

Para acessar uma variável específica (ex: `amsua`), faça:

```
[11]: print('Variável: ', varName)

gdf_list[tidx].obsInfo[varName]
```

Variável: amsua

```
[11]:          lat          lon          elev  nchan          time  iuse  idqc  \
SatId points
n19  0      -87.446198    6.644900  2647.032227    1.0  2.885000    1.0    8.0
      1      -87.446198    6.644900  2647.032227    2.0  2.885000    1.0    8.0
      2      -87.446198    6.644900  2647.032227    3.0  2.885000    1.0    8.0
      3      -87.446198    6.644900  2647.032227    4.0  2.885000    1.0    8.0
      4      -87.446198    6.644900  2647.032227    5.0  2.885000    1.0    8.0
...
      137830  56.754601   336.250885    0.055578   11.0 -1.522778    1.0    0.0
      137831  56.754601   336.250885    0.055578   12.0 -1.522778    1.0    0.0
      137832  56.754601   336.250885    0.055578   13.0 -1.522778    1.0    0.0
      137833  56.754601   336.250885    0.055578   14.0 -1.522778    4.0    0.0
      137834  56.754601   336.250885    0.055578   15.0 -1.522778    1.0   51.0
```

```
          inverr          oer          obs          omf  omf_nobc          emiss  \
SatId points
n19  0      0.000000          NaN  187.330002    2.591137   -7.817700    0.813564
      1      0.000000          NaN  189.190002    4.114823   -3.598879    0.801515
      2      0.000000          NaN  211.460007   -1.666820    2.795111    0.791668
      3      0.000000          NaN  225.820007    0.495127   -0.318058    0.791438
      4      0.000000          NaN  229.089996    0.241357   -1.846274    0.790830
...
      137830  2.499857    0.400023  219.949997    0.632804   -0.609745    0.856901
      137831  1.817697    0.550150  219.039993    0.488474   -0.474023    0.856901
      137832  1.248405    0.801043  217.449997   -1.116789   -1.843264    0.856901
      137833  0.248456    4.025372  219.979996   -0.762485   -0.762485    0.856901
      137834  0.000000          NaN  229.039993  -23.291994  -41.154434    0.950000
```

```
          oma  oma_nobc          imp          dfs  \
SatId points
n19  0      2.803686   -7.640147          NaN          NaN
      1      4.388880   -3.414203          NaN          NaN
      2     -1.364057    3.049402          NaN          NaN
      3      0.594996   -0.230114          NaN          NaN
      4      0.297319   -1.806047          NaN          NaN
...
      137830  0.451392   -0.792813   -0.491687   -0.286979
      137831  0.409210   -0.554484   -0.129336   -0.070378
      137832  -0.789088   -1.543222   -0.779681   -0.456870
      137833  0.452604    0.452604   -0.093540   -0.230162
      137834 -23.673958  -41.430882          NaN          NaN
```

```
          geometry
SatId points
n19  0      POINT (6.64490 -87.44620)
      1      POINT (6.64490 -87.44620)
      2      POINT (6.64490 -87.44620)
      3      POINT (6.64490 -87.44620)
      4      POINT (6.64490 -87.44620)
...
      137830 POINT (-23.74915 56.75460)
```

```

137831 POINT (-23.74915 56.75460)
137832 POINT (-23.74915 56.75460)
137833 POINT (-23.74915 56.75460)
137834 POINT (-23.74915 56.75460)

```

[137835 rows x 18 columns]

Para acessar variável e tipo específicos (ex: `amsua` do tipo `n19`), faça:

```

[12]: print('Variável: ', varName, ' e Tipo: ', varType)

gdf_list[tidx].obsInfo[varName].loc[varType]

```

Variável: `amsua` e Tipo: `n19`

```

[12]:      lat      lon      elev  nchan      time  iuse  idqc  \
points
0      -87.446198    6.644900  2647.032227    1.0  2.885000    1.0    8.0
1      -87.446198    6.644900  2647.032227    2.0  2.885000    1.0    8.0
2      -87.446198    6.644900  2647.032227    3.0  2.885000    1.0    8.0
3      -87.446198    6.644900  2647.032227    4.0  2.885000    1.0    8.0
4      -87.446198    6.644900  2647.032227    5.0  2.885000    1.0    8.0
...      ...      ...      ...      ...      ...      ...      ...
137830  56.754601   336.250885    0.055578   11.0 -1.522778    1.0    0.0
137831  56.754601   336.250885    0.055578   12.0 -1.522778    1.0    0.0
137832  56.754601   336.250885    0.055578   13.0 -1.522778    1.0    0.0
137833  56.754601   336.250885    0.055578   14.0 -1.522778    4.0    0.0
137834  56.754601   336.250885    0.055578   15.0 -1.522778    1.0   51.0

```

```

      inverr      oer      obs      omf  omf_nobc      emiss  \
points
0      0.000000      NaN  187.330002   2.591137  -7.817700   0.813564
1      0.000000      NaN  189.190002   4.114823  -3.598879   0.801515
2      0.000000      NaN  211.460007  -1.666820   2.795111   0.791668
3      0.000000      NaN  225.820007   0.495127  -0.318058   0.791438
4      0.000000      NaN  229.089996   0.241357  -1.846274   0.790830
...      ...      ...      ...      ...      ...      ...
137830  2.499857   0.400023  219.949997   0.632804  -0.609745   0.856901
137831  1.817697   0.550150  219.039993   0.488474  -0.474023   0.856901
137832  1.248405   0.801043  217.449997  -1.116789  -1.843264   0.856901
137833  0.248456   4.025372  219.979996  -0.762485  -0.762485   0.856901
137834  0.000000      NaN  229.039993 -23.291994 -41.154434   0.950000

```

```

      oma  oma_nobc      imp      dfs      geometry
points
0      2.803686  -7.640147      NaN      NaN  POINT (6.64490 -87.44620)
1      4.388880  -3.414203      NaN      NaN  POINT (6.64490 -87.44620)
2     -1.364057   3.049402      NaN      NaN  POINT (6.64490 -87.44620)
3      0.594996  -0.230114      NaN      NaN  POINT (6.64490 -87.44620)
4      0.297319  -1.806047      NaN      NaN  POINT (6.64490 -87.44620)
...      ...      ...      ...      ...      ...
137830  0.451392  -0.792813 -0.491687 -0.286979  POINT (-23.74915 56.75460)
137831  0.409210  -0.554484 -0.129336 -0.070378  POINT (-23.74915 56.75460)
137832 -0.789088  -1.543222 -0.779681 -0.456870  POINT (-23.74915 56.75460)
137833  0.452604  0.452604 -0.093540 -0.230162  POINT (-23.74915 56.75460)
137834 -23.673958 -41.430882      NaN      NaN  POINT (-23.74915 56.75460)

```

[137835 rows x 18 columns]

Para acessar o valor da observação, utilize o método `obs`:

```
[13]: print('Variável: ', varName, ' e Tipo: ', varType)

      gdf_list[tidx].obsInfo[varName].loc[varType].obs
```

Variável: amsua e Tipo: n19

```
[13]: points
0      187.330002
1      189.190002
2      211.460007
3      225.820007
4      229.089996
...
137830  219.949997
137831  219.039993
137832  217.449997
137833  219.979996
137834  229.039993
Name: obs, Length: 137835, dtype: float32
```

Para acessar o valor do controle de qualidade da observação em um canal específico, utilize o método `idqc` com a instrução `mask` na função `query`:

```
[14]: mask = 'nchan == 2'
      gdf_list[tidx].obsInfo[varName].query(mask).loc[varType].idqc
```

```
[14]: points
1      8.0
16     8.0
31    51.0
46    51.0
61    51.0
...
137761  51.0
137776  51.0
137791  51.0
137806  51.0
137821  51.0
Name: idqc, Length: 9189, dtype: float32
```

1.4 Utilização da classe `plot_diag`

1.4.1 Distribuição espacial

A utilização das funções da classe `plot_diag` que foram alteradas/construídas para diagnóstico de dados de **radiância** é apresentada abaixo, junto com os comandos para gerar diversos tipos de figuras.

Gerando uma figura com os valores das observações (`param='obs'`) para a variável e tipo selecionados:

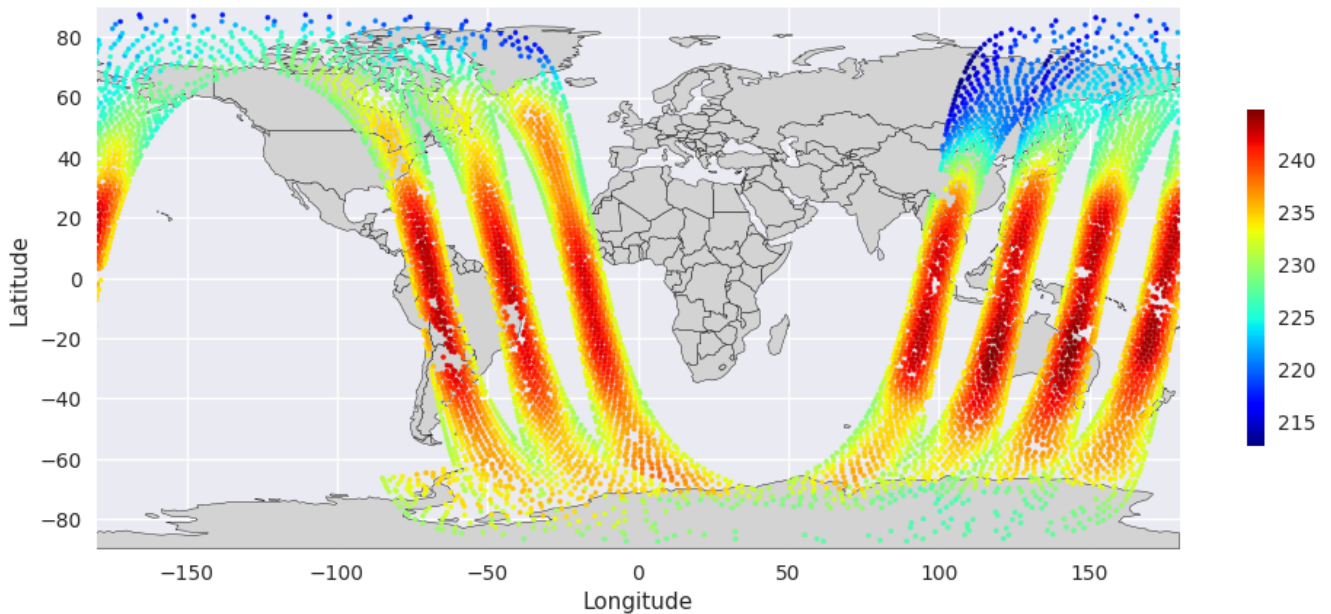
```
[15]: param = 'obs'

gd.plot_diag.plot(gdf_list[tidx],
                  varName=varName,
                  varType=varType,
                  param=param,
```

```
mask="(nchan==" + str(channel) + ") & (iuse >= 1 & idqc == 0)",
markersize=4.80,
legend='true')
```

/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.
path=gpd.datasets.get_path('naturalearth_lowres')

[15]: <Axes: xlabel='Longitude', ylabel='Latitude'>



Abaixo, um exemplo alterando a escala de cores do colormap (`minVal` e `maxVal`), inserido informações sobre canal, data, sensor e satélite (aqui utiliza a função `getVarInfo` da biblioteca `datasources`) e salvando a figura com o nome proposto em `figname`:

```
[16]: param = 'obs'
minVal = 200.0
maxVal = 250.0

forplot = 'Channel =' + str(channel)
varInfo = getVarInfo(varType, varName, 'instrument')
if varInfo is not None:
    instrument_title = varInfo
else:
    instrument_title = str(varName) + '-' + str(varType) + ' | ' + 'Unknown instrument'

ax = gd.plot_diag.plot(gdf_list[tidx],
                        varName=varName,
                        varType=varType,
                        param=param,
                        minVal=minVal,
                        maxVal=maxVal,
                        mask="(nchan==" + str(channel) + ") & (iuse >= 1 & idqc == 0)",
                        markersize=4.80,
                        legend='true')
```

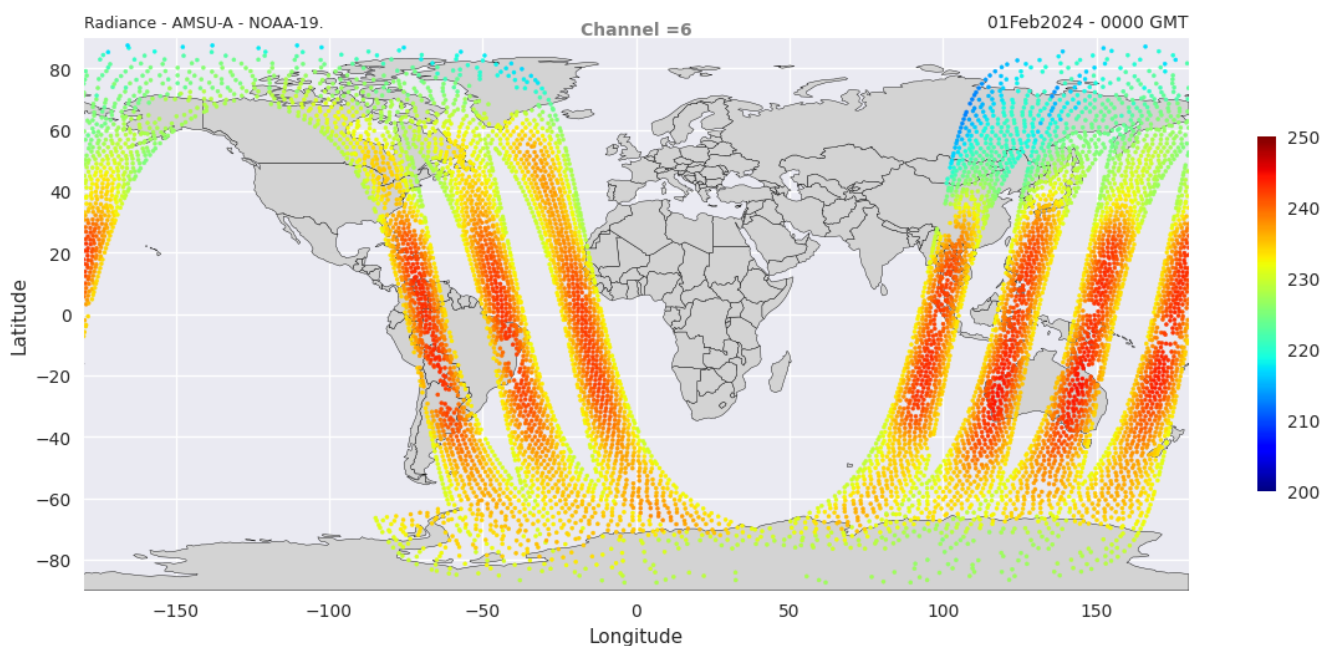
```

if ax != None:
    datefmt = dates[tidx]
    date = datetime.strptime(str(datefmt), "%Y%m%d%H")
    date_title = str(date.strftime("%d%b%Y - %H%M")) + ' GMT'
    plt.title(date_title, loc='right', fontsize=10)
    plt.title(instrument_title, loc='left', fontsize=9)
    plt.annotate(forplot, xy=(0.45, 1.015), xytext=(0, 0), xycoords='axes fraction',
    ↪textcoords='offset points',
    color='gray', fontweight='bold', fontsize='10', horizontalalignment='left',
    ↪verticalalignment='center')

    figname=varName+"_"+str(varType)+"_"+param+"_"+str(channel)+"_"+dates[tidx]+"_"+str(channel)+"_plot.png"
    plt.tight_layout()
    plt.savefig(figname)

```

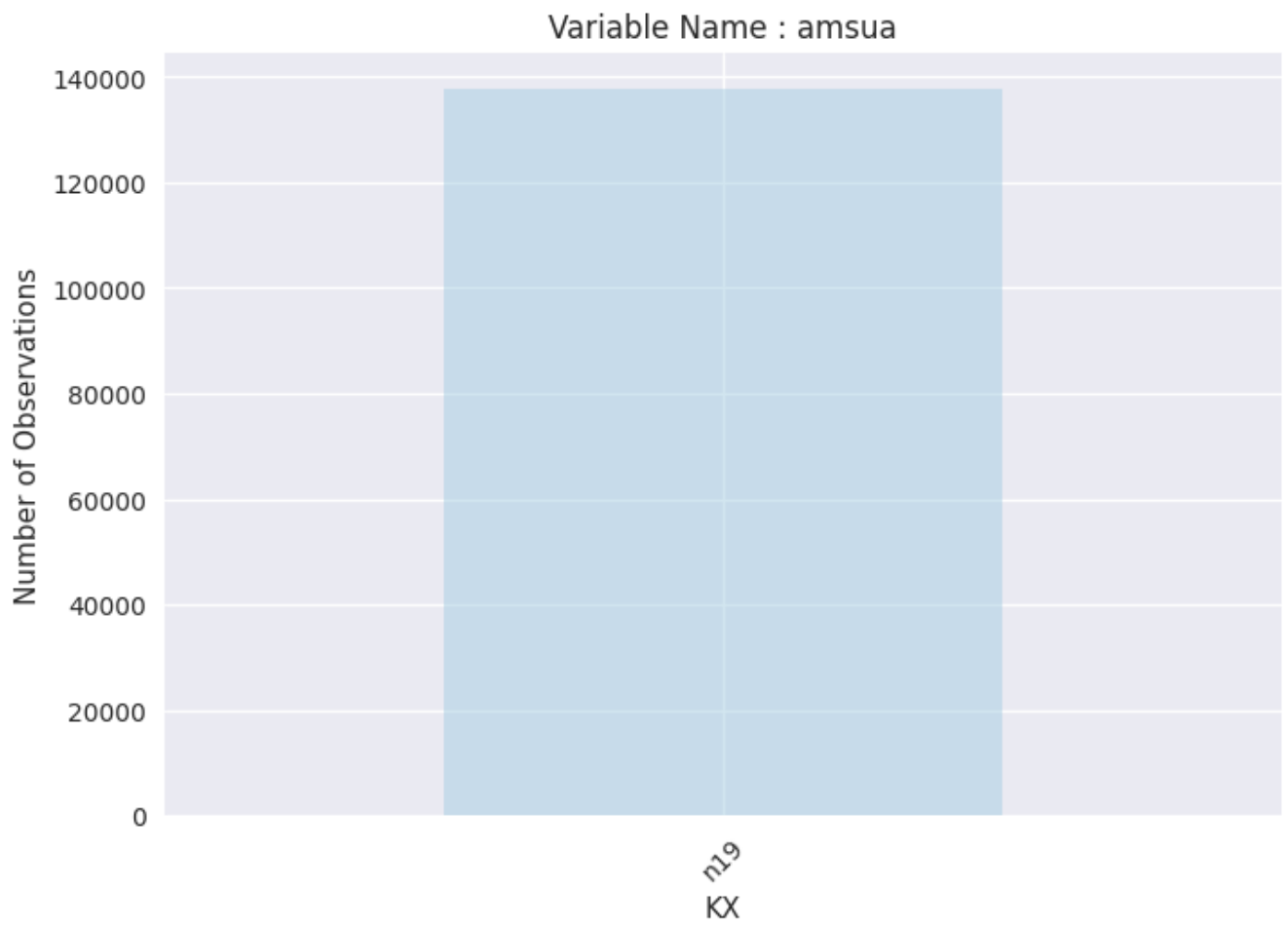
/home/radiancia/DiagRadi-V2_issue45/gsideag/__main__.py:93: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.
 path=gpd.datasets.get_path('naturalearth_lowres')



1.4.2 Histograma

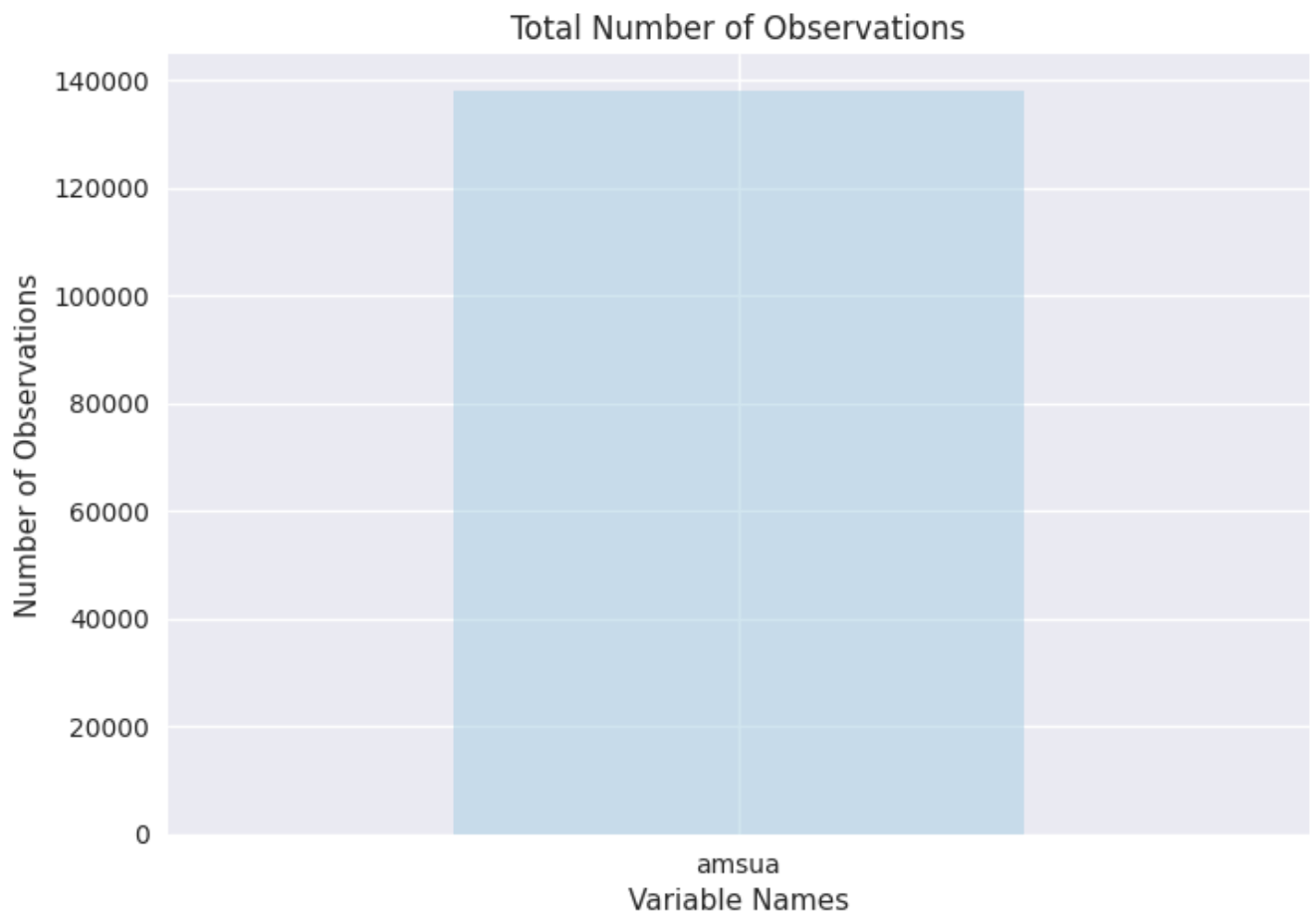
As funções `pcount()` e `vcount()` da classe `plot_diag` também podem ser utilizadas para obter um histograma com a contagem do número de observações para a variável, porém, no caso da **radiância**, como há apenas os dados de uma variável (sensor) e um tipo (satélite) nos arquivos, ambas as funções geram o mesmo histograma com uma única coluna:

```
[17]: gd.plot_diag.pcount(gdf_list[tidx], varName)
```



Utilize a função `vcount()` da classe `plot_diag` para obter um histograma com a contagem do número de observações para todos os tipos de variáveis:

```
[18]: gd.plot_diag.vcount(gdf_list[tidx])
```



1.4.3 Série temporal

A seguir são apresentadas as opções de figuras com a função `time_series_radi()` (nova função), incluída na classe `plot_diag`. Inicialmente é gerada uma figura com os parâmetros já fixados nesta seção. Em seguida são apresentadas figuras alterando o parâmetro `channel`.

Plotando uma série temporal do OmA e OmF para os dados do canal setado na célula de definição das variáveis:

```
[19]: gd.plot_diag.time_series_radi(gdf_list,
                                     varName=varName,
                                     varType=varType,
                                     mask='idqc==0 & iuse>=1',
                                     dateIni=dateIni,
                                     dateFin=dateFin,
                                     nHour=nHour,
                                     vminOMA=vminOMA,
                                     vmaxOMA=vmaxOMA,
                                     vminSTD=vminSTD,
                                     vmaxSTD=vmaxSTD,
                                     channel=channel,
                                     Clean=False)
```

```
=====
=====
Variable: amsua || type: n19 || Radiance - AMSU-A - NOAA-19. || check:
OmF
```

```
=====
=====

Preparing data for: 2024-02-01:00 - Channel de radiancia: 6
Preparing data for: 2024-02-01:06 - Channel de radiancia: 6
Preparing data for: 2024-02-01:12 - Channel de radiancia: 6
Preparing data for: 2024-02-01:18 - Channel de radiancia: 6
Preparing data for: 2024-02-02:00 - Channel de radiancia: 6
Preparing data for: 2024-02-02:06 - Channel de radiancia: 6
Preparing data for: 2024-02-02:12 - Channel de radiancia: 6
Preparing data for: 2024-02-02:18 - Channel de radiancia: 6

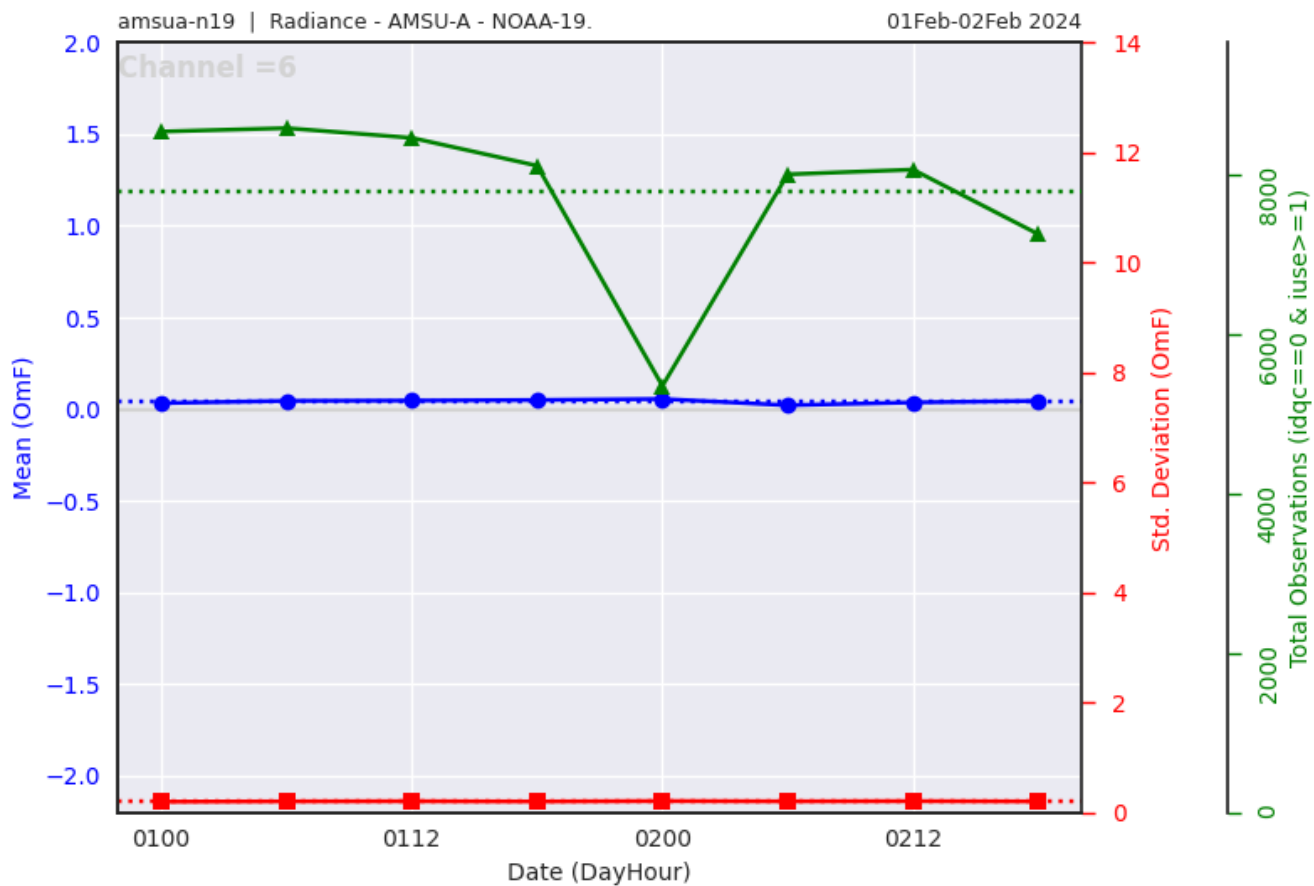
=====
=====
```

```
channels = [6]
Calculating for 2024-02-01:00
Calculating for 2024-02-01:06
Calculating for 2024-02-01:12
Calculating for 2024-02-01:18
Calculating for 2024-02-02:00
Calculating for 2024-02-02:06
Calculating for 2024-02-02:12
Calculating for 2024-02-02:18

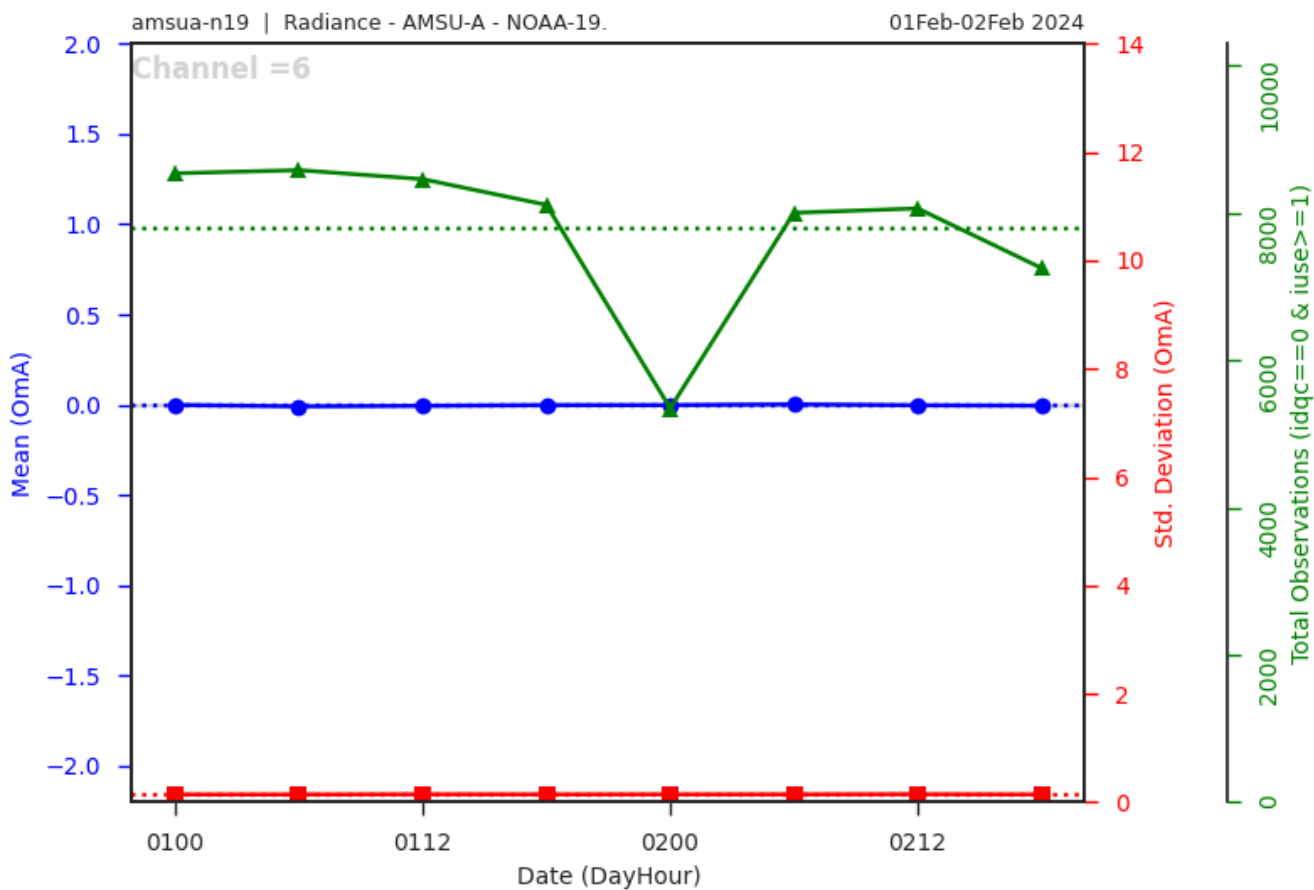
=====
=====
```

```
Making Graphics...
Done!
```

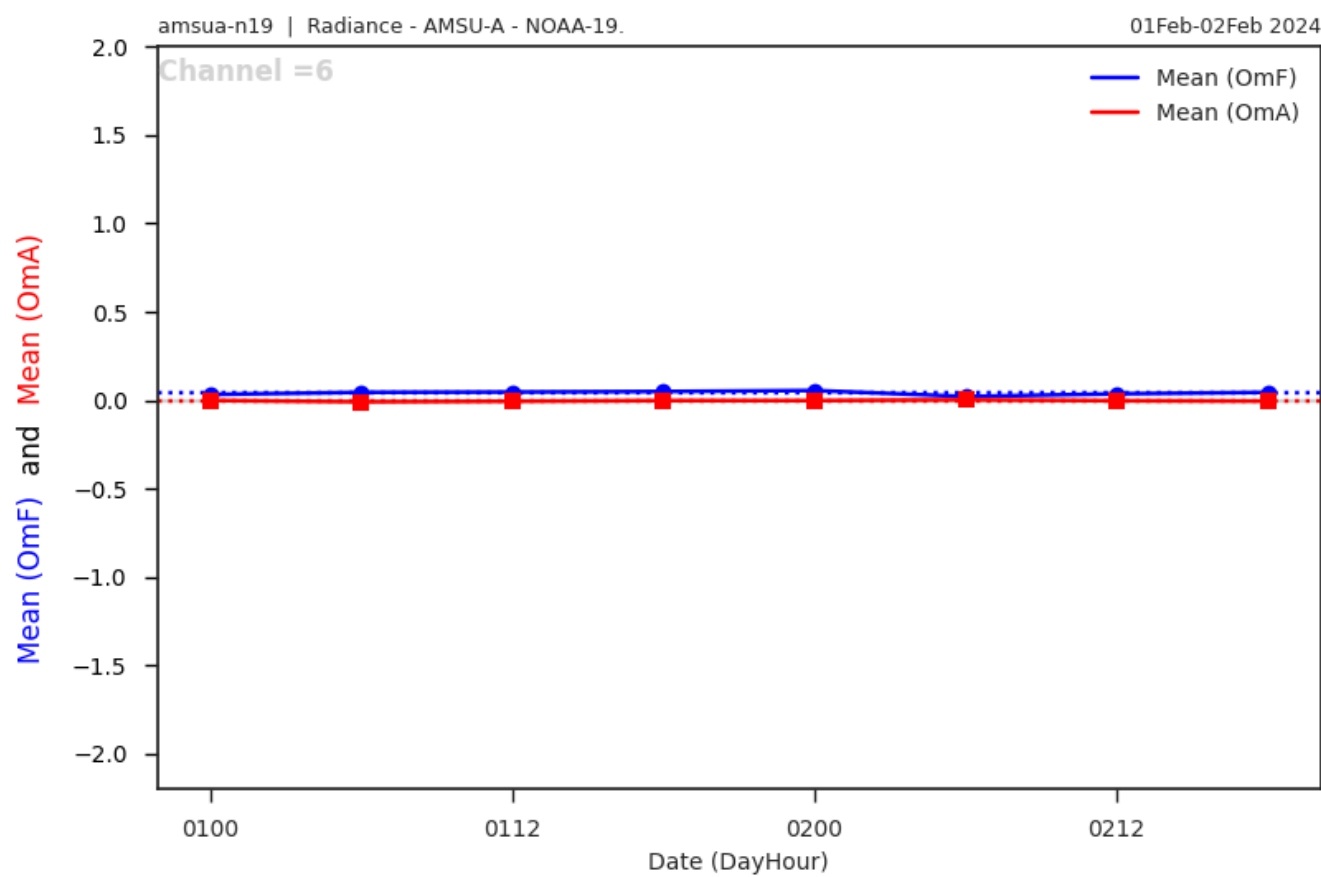
```
<Figure size 600x400 with 0 Axes>
```



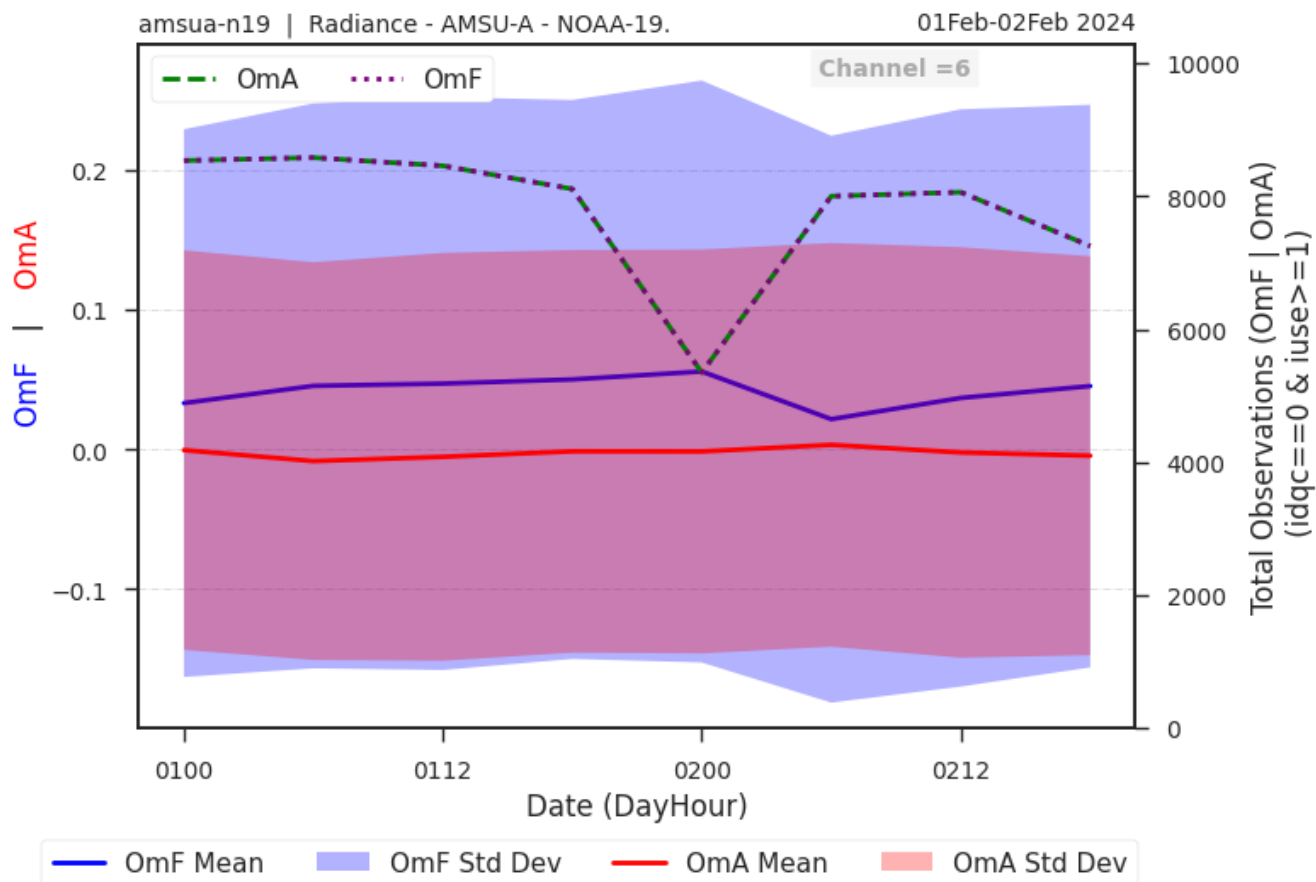
<Figure size 600x400 with 0 Axes>



<Figure size 600x400 with 0 Axes>



<Figure size 600x400 with 0 Axes>



No caso anterior, o parâmetro `channel` estava fixo no canal 6. Os valores de `vminOMA`, `vmaxOMA`, `vminSTD` e `vmaxSTD`, fixados inicialmente, são alterados para melhorar a visualização da série temporal.

```
[20]: vminOMA = -0.5
vmaxOMA = 0.5
vminSTD = 0.0
vmaxSTD = 1.0

gd.plot_diag.time_series_radi(gdf_list,
                               varName=varName,
                               varType=varType,
                               mask='idqc==0 & iuse>=1',
                               dateIni=dateIni,
                               dateFin=dateFin,
                               nHour=nHour,
                               vminOMA=vminOMA,
                               vmaxOMA=vmaxOMA,
                               vminSTD=vminSTD,
                               vmaxSTD=vmaxSTD,
                               channel=channel,
                               Clean=False)
```

```
=====
Variable: amsua || type: n19 || Radiance - AMSU-A - NOAA-19. || check:
OmF
=====
```

=====

Preparing data for: 2024-02-01:00 - Channel de radiancia: 6
Preparing data for: 2024-02-01:06 - Channel de radiancia: 6
Preparing data for: 2024-02-01:12 - Channel de radiancia: 6
Preparing data for: 2024-02-01:18 - Channel de radiancia: 6
Preparing data for: 2024-02-02:00 - Channel de radiancia: 6
Preparing data for: 2024-02-02:06 - Channel de radiancia: 6
Preparing data for: 2024-02-02:12 - Channel de radiancia: 6
Preparing data for: 2024-02-02:18 - Channel de radiancia: 6

=====

=====

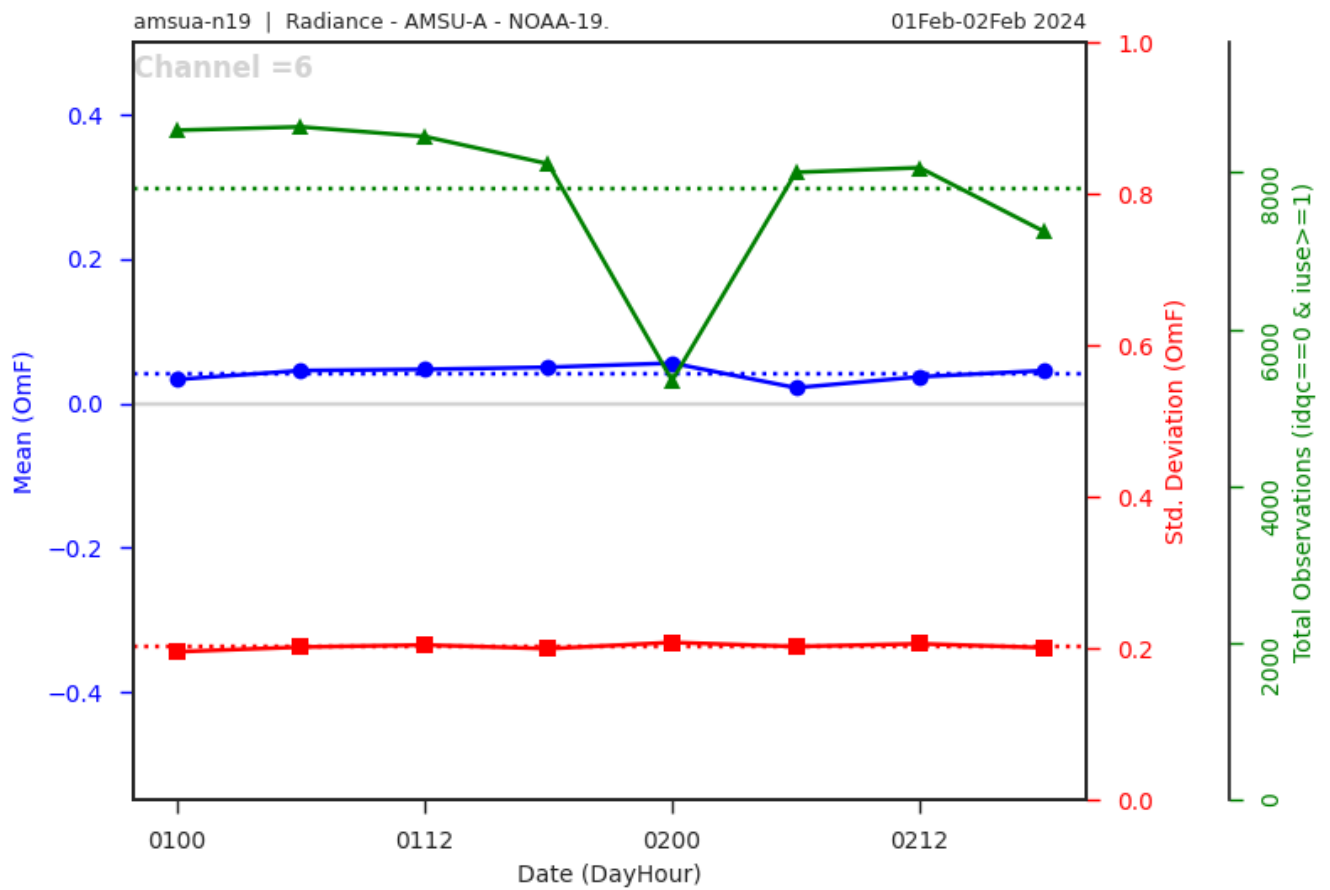
channels = [6]
Calculating for 2024-02-01:00
Calculating for 2024-02-01:06
Calculating for 2024-02-01:12
Calculating for 2024-02-01:18
Calculating for 2024-02-02:00
Calculating for 2024-02-02:06
Calculating for 2024-02-02:12
Calculating for 2024-02-02:18

=====

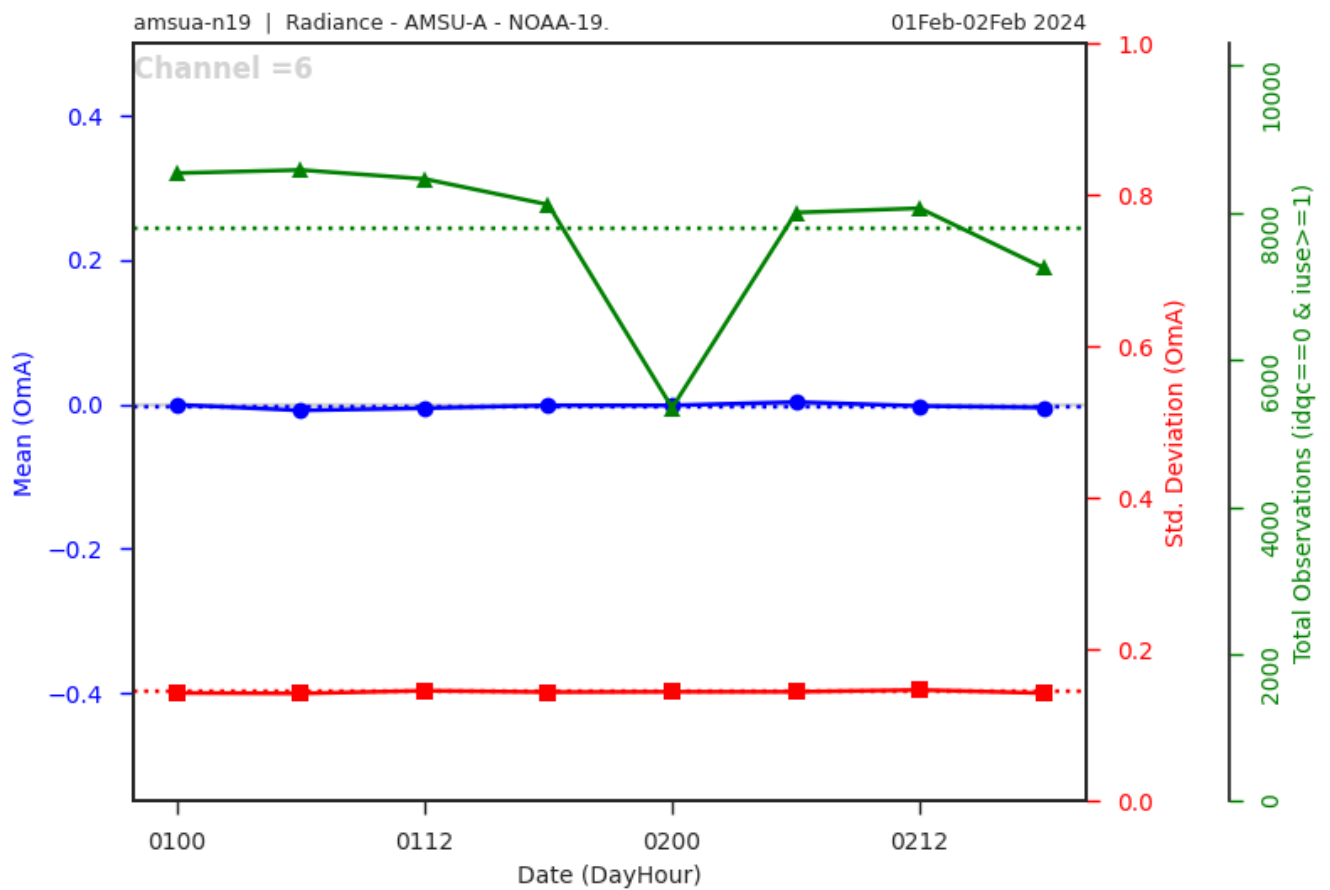
=====

Making Graphics...
Done!

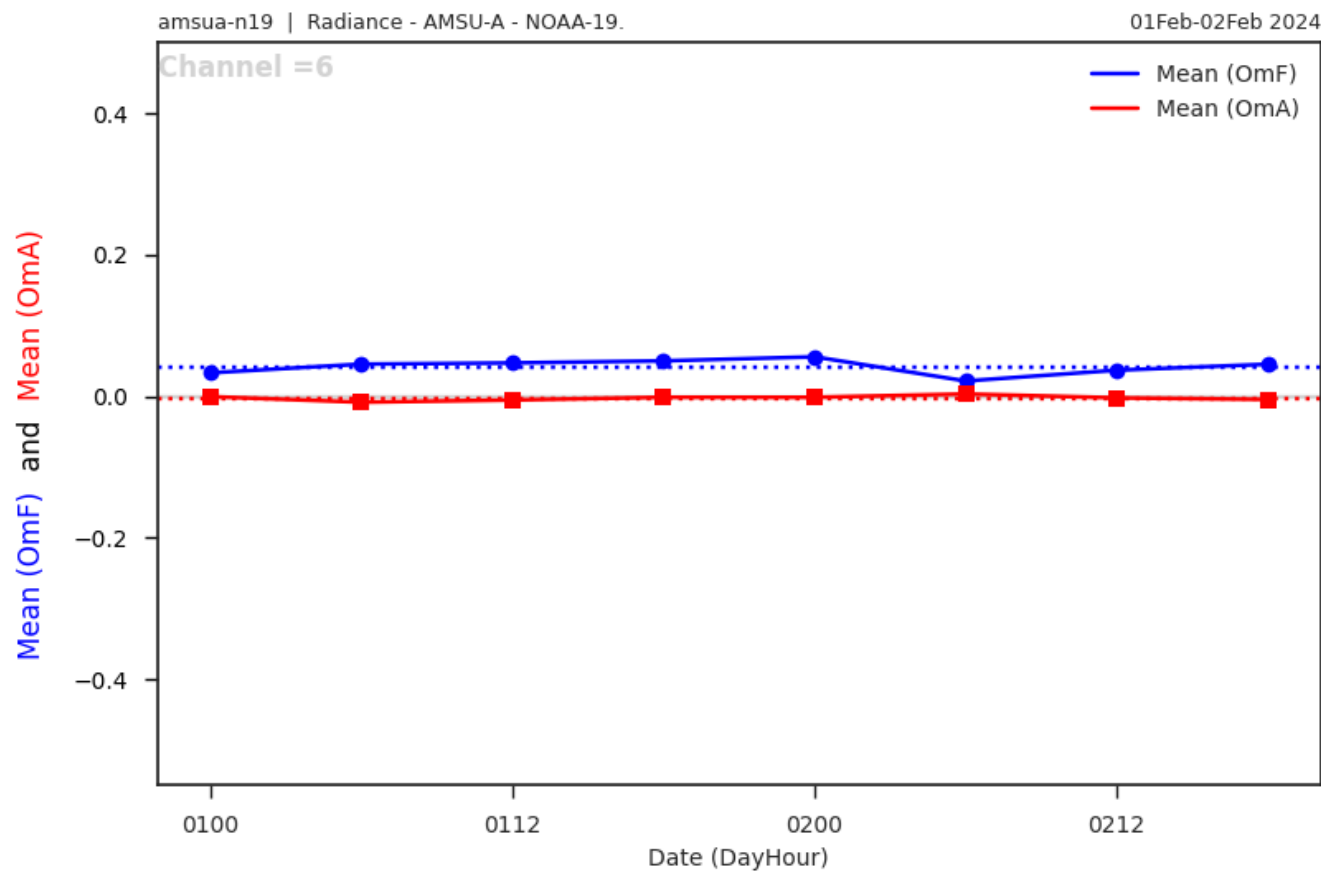
<Figure size 600x400 with 0 Axes>



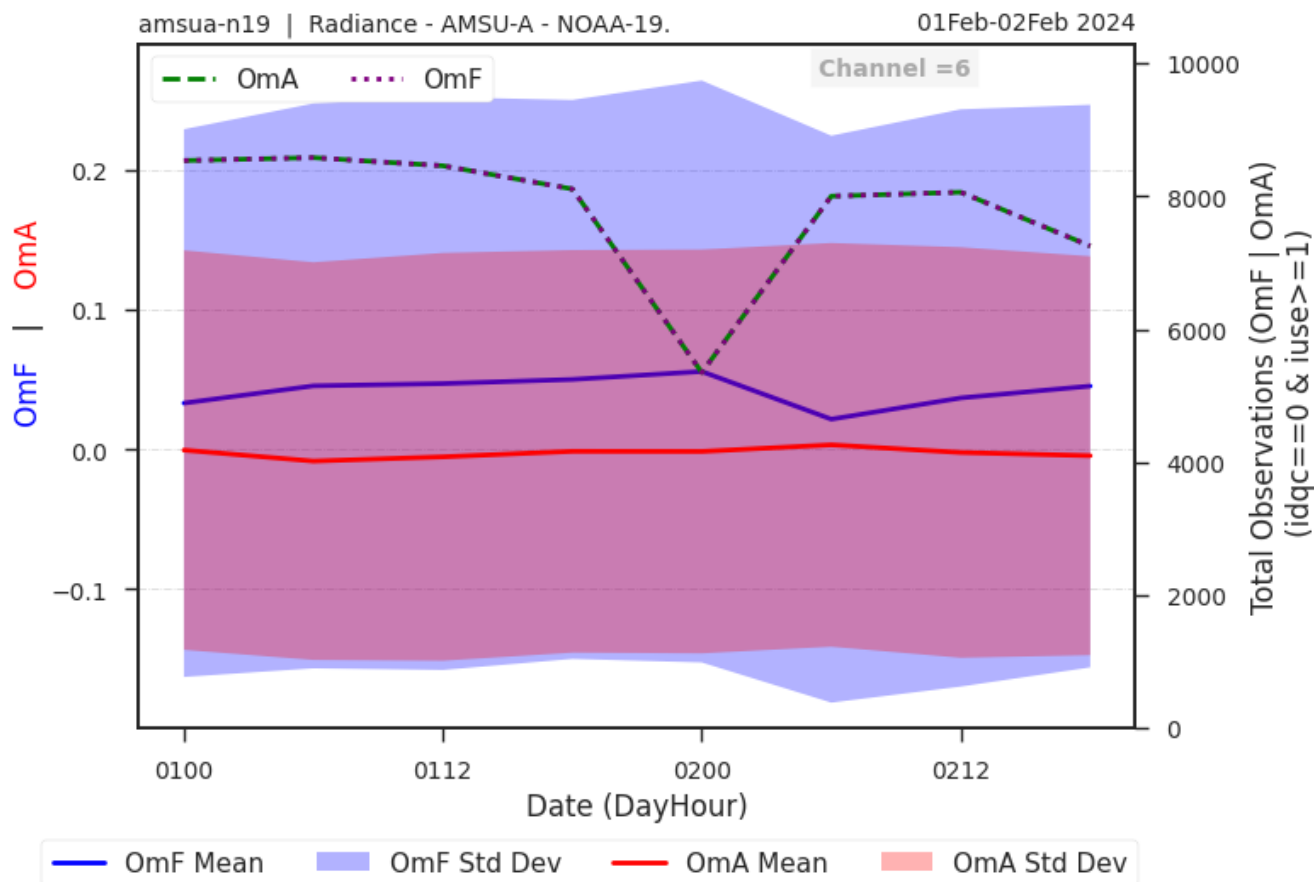
<Figure size 600x400 with 0 Axes>



<Figure size 600x400 with 0 Axes>



<Figure size 600x400 with 0 Axes>



Agora, a variável `channel` é alterada para considerar mais de um canal. A opção `channel = None` busca os dados em cada canal existente nos arquivos e faz um diagrama de *Hovmöller* com as médias, desvios padrão e quantidade de dados assimilados em todos os canais (observe que os valores no eixo y (ordenada) identifica todos os canais de 1 à 15). Quando `channel` é uma lista específica de canais, indicada pelo usuário, o diagrama de *Hovmöller* é construído apenas com os canais indicados pelo usuário.

Dessa forma, pode-se utilizar o parâmetro `channels=[6,10,11,12,13,14]` para produzir uma série temporal para mais de um canal:

```
[21]: channels = [6, 10, 11, 12, 13, 14]

gd.plot_diag.time_series_radi(gdf_list,
                              varName=varName,
                              varType=varType,
                              mask='idqc==0 & iuse>=1',
                              dateIni=dateIni,
                              dateFin=dateFin,
                              nHour=nHour,
                              vminOMA=vminOMA,
                              vmaxOMA=vmaxOMA,
                              vminSTD=vminSTD,
                              vmaxSTD=vmaxSTD,
                              channel=channels,
                              Clean=False)
```

```
=====
Variable: amsua || type: n19 || Radiance - AMSU-A - NOAA-19. || check:
```

OmF

=====

Preparing data for: Canais de radiancia2024-02-01:00
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-01:06
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-01:12
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-01:18
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-02:00
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-02:06
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-02:12
Channels: [6, 10, 11, 12, 13, 14]

Preparing data for: Canais de radiancia2024-02-02:18
Channels: [6, 10, 11, 12, 13, 14]

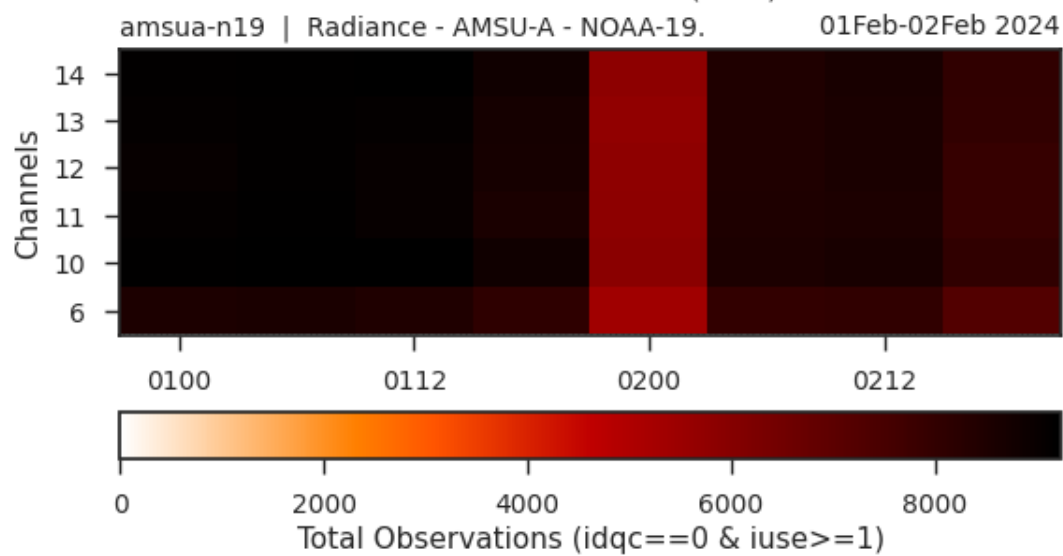
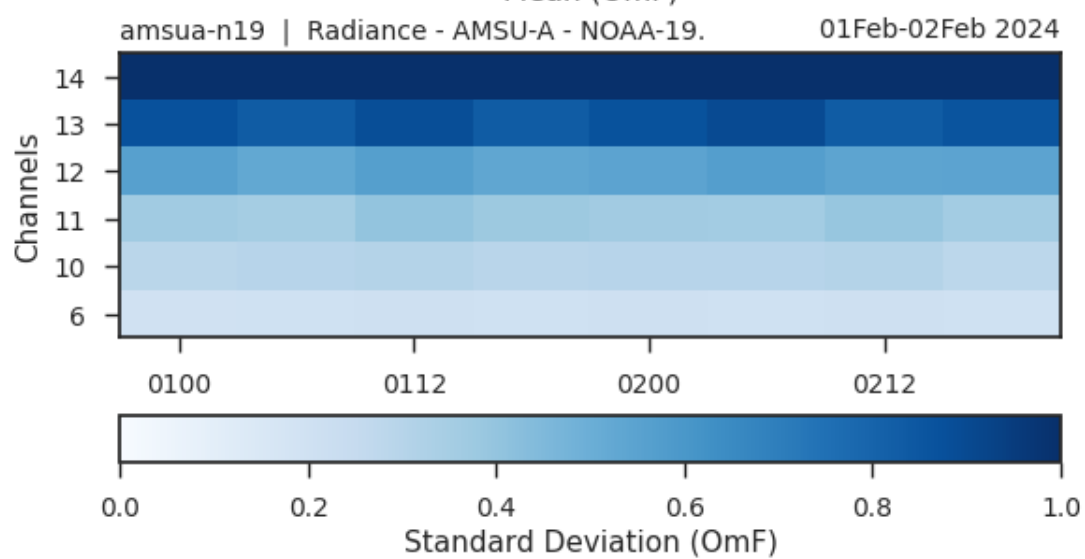
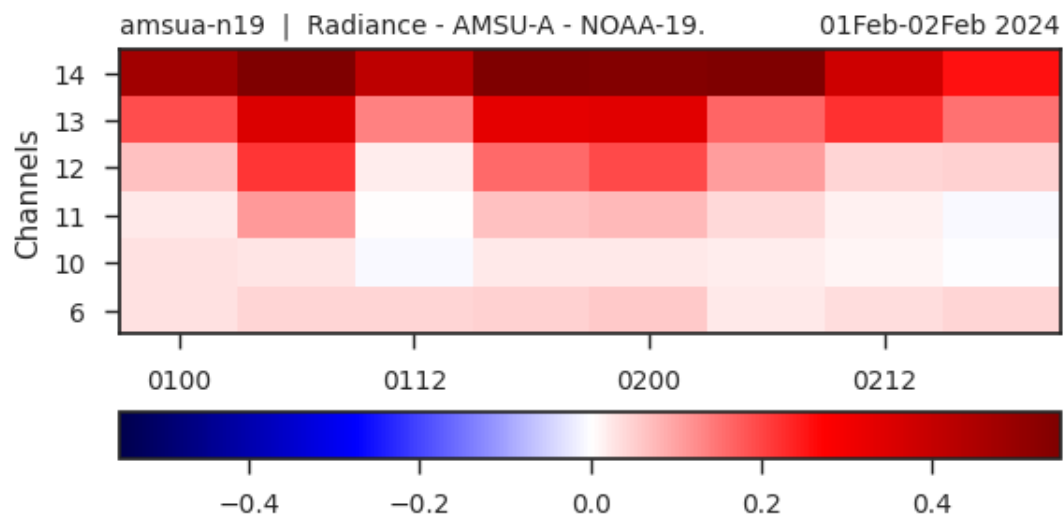
=====

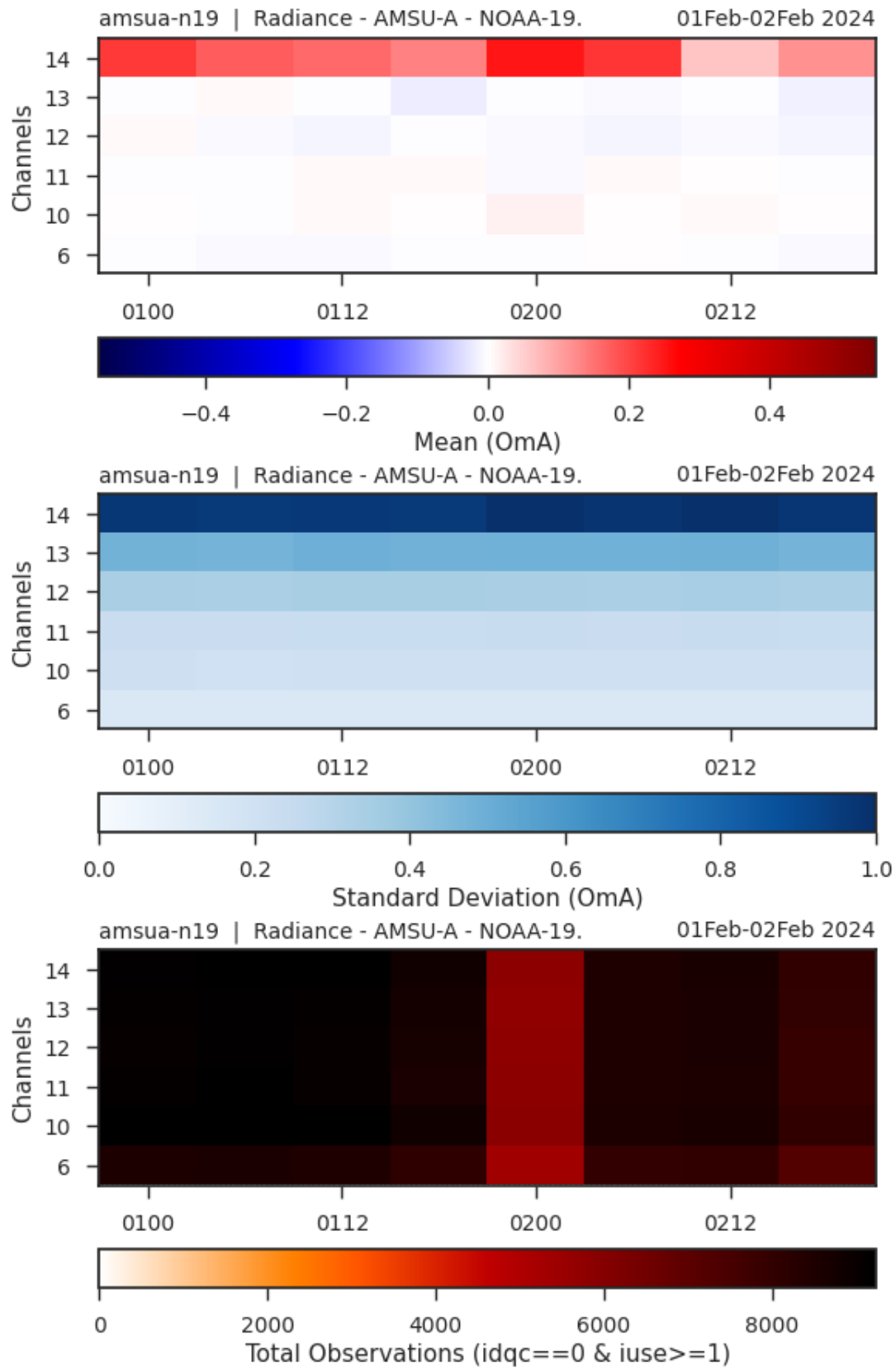
channels = [6, 10, 11, 12, 13, 14]

Calculating for 2024-02-01:00
Calculating for 2024-02-01:06
Calculating for 2024-02-01:12
Calculating for 2024-02-01:18
Calculating for 2024-02-02:00
Calculating for 2024-02-02:06
Calculating for 2024-02-02:12
Calculating for 2024-02-02:18

=====

Making Graphics...
Done!





Considerando todos os canais, ou seja, `channel = None`:

```
[22]: channel = None

gd.plot_diag.time_series_radi(gdf_list,
```

```
varName=varName,  
varType=varType,  
mask='idqc==0 & iuse>=1',  
dateIni=dateIni,  
dateFin=dateFin,  
nHour=nHour,  
vminOMA=vminOMA,  
vmaxOMA=vmaxOMA,  
vminSTD=vminSTD,  
vmaxSTD=vmaxSTD,  
channel=channel,  
Clean=False)
```

```
=====
```

```
Variable: amsua || type: n19 || Radiance - AMSU-A - NOAA-19. || check:  
OmF
```

```
=====
```

```
=====
```

```
Preparing data for: Canais de radiancia2024-02-01:00  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-01:06  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-01:12  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-01:18  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-02:00  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-02:06  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-02:12  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
Preparing data for: Canais de radiancia2024-02-02:18  
Channels: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

```
=====
```

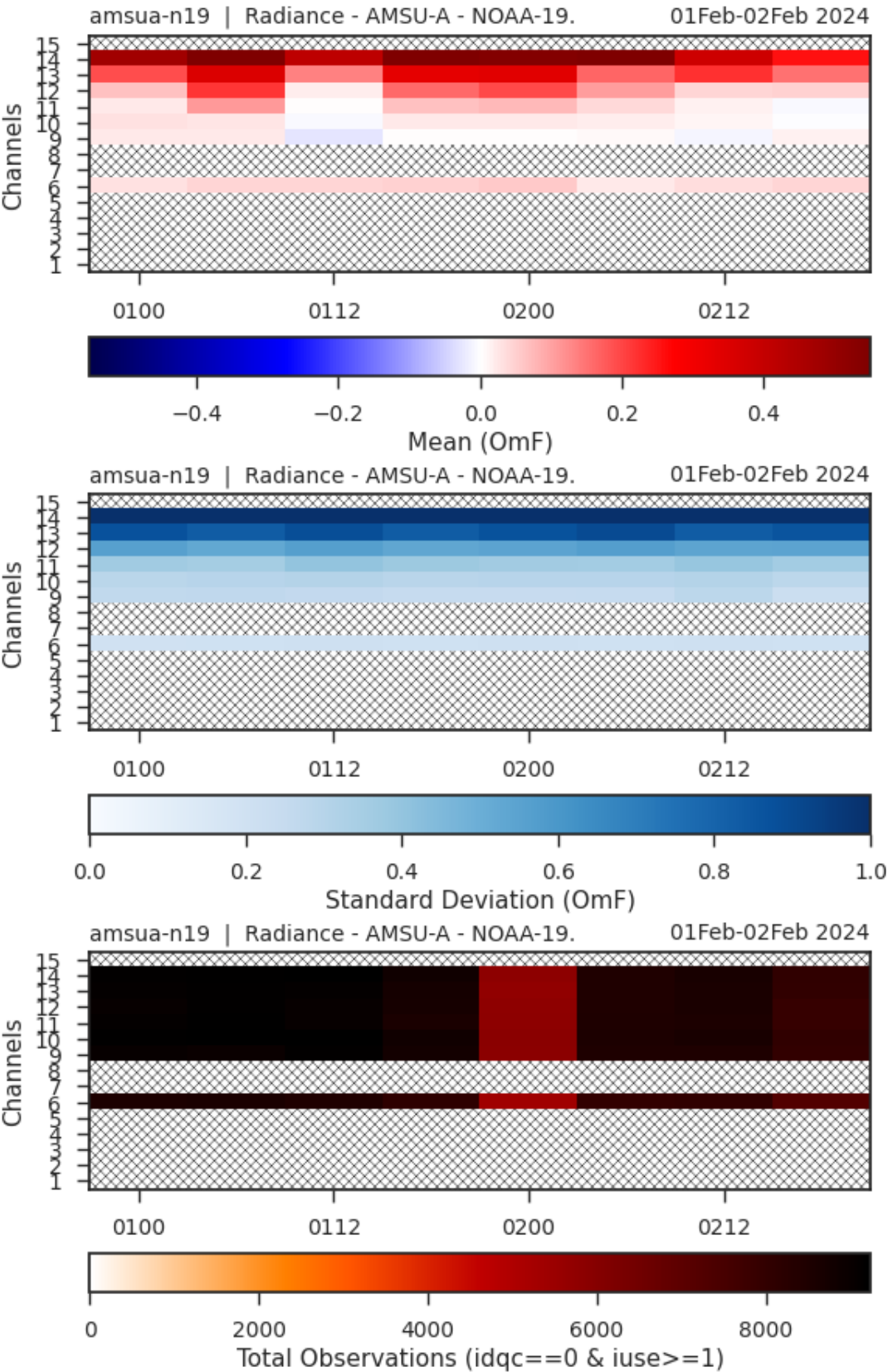
```
=====
```

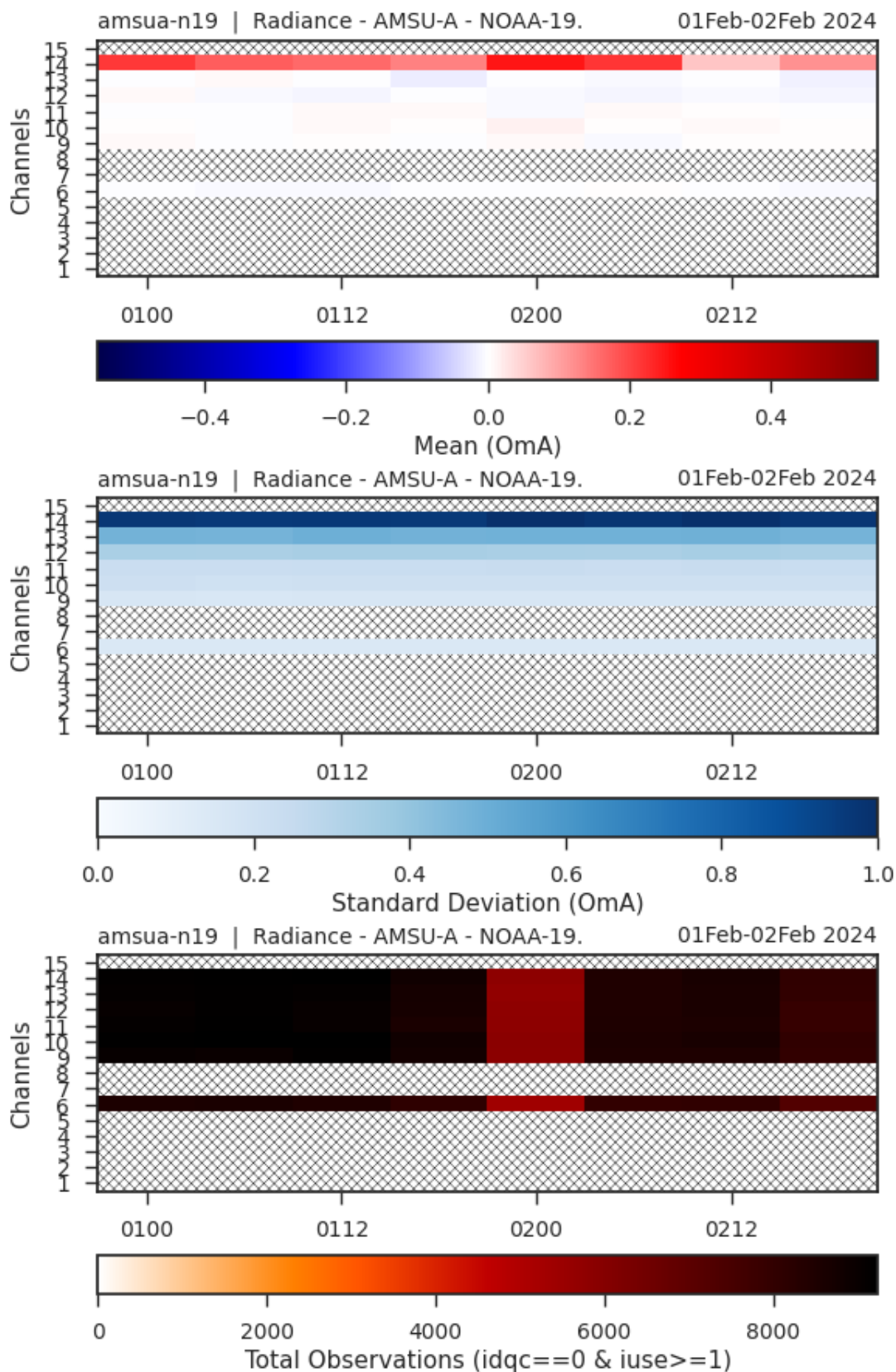
```
channels = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]  
Calculating for 2024-02-01:00  
Calculating for 2024-02-01:06  
Calculating for 2024-02-01:12  
Calculating for 2024-02-01:18  
Calculating for 2024-02-02:00  
Calculating for 2024-02-02:06  
Calculating for 2024-02-02:12
```

Calculating for 2024-02-02:18

=====

Making Graphics...
Done!





1.4.4 STATCOUNT

A seguir são apresentadas as opções de figuras obtidas pela função `statcount()`. A função `statcount()` fornece dois tipos de figuras: com `figMap=True` é possível gerar uma sequência de imagens com os pontos geográficos das observações assimiladas, rejeitadas e monitoradas em um período de tempo `dateIni` à `dateFin` com intervalo de tempo `nHour`. Uma

outra figura contendo uma série temporal da quantidade de dados assimilados, rejeitados e monitorados é gerada se `figTS=True`. Quando os arquivos utilizados são de **radiância**, deve-se atribuir a variável `channel` o número do canal desejado (`channel=6`), enquanto que, se os dados analisados forem convencionais a variável `channel` deve receber `None` (`channel=None`).

Nos dados de **radiância** a classificação dos dados em assimilado, rejeitado e monitorado é realizada considerando dois tipos de observações monitoradas:

- 1. monitorado-assimilado: observações que são monitoradas, `iuse = -1`, que poderiam ser assimiladas pois `idqc = 0`.
- 2. monitorado-rejeitado: observações que são monitoradas, `iuse = -1`, que poderiam ser rejeitadas pois `idqc != 0`.

A tabela abaixo detalha como a classificação dos dados de **radiância** é realizada.

	idqc	iuse
Assimilated	<code>== 0</code>	<code>>= 1</code>
Monitored-assimilated	<code>== 0</code>	<code>>= -1 and < 1</code>
Monitored-rejected	<code>!= 0</code>	<code>>= -1 and < 1</code>
Rejected	<code>!= 0</code>	<code>>= 1</code>

A seguir um exemplo que gera uma sequência de figuras com a geolocalização dos dados de **radiância** assimilados, rejeitados e monitorados (`figMap=True`) em um intervalo de tempo dentro do período de dados carregado no início do notebook.

```
[23]: channel = 6
dateIni = '2024020100'
dateFin = '2024020118'

gd.plot_diag.statcount(gdf_list,
                        varName=varName,
                        varType=varType,
                        noiqc=False,
                        dateIni=dateIni,
                        dateFin=dateFin,
                        nHour=nHour,
                        channel=channel,
                        figTS=False,
                        figMap=True,
                        markersize=4.80)
```

```
/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You
can get the original 'naturalearth_lowres' data from
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  path=gpd.datasets.get_path('naturalearth_lowres')

channel 6 not monitored on the date --> 2024-02-01:00

/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You
can get the original 'naturalearth_lowres' data from
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  path=gpd.datasets.get_path('naturalearth_lowres')

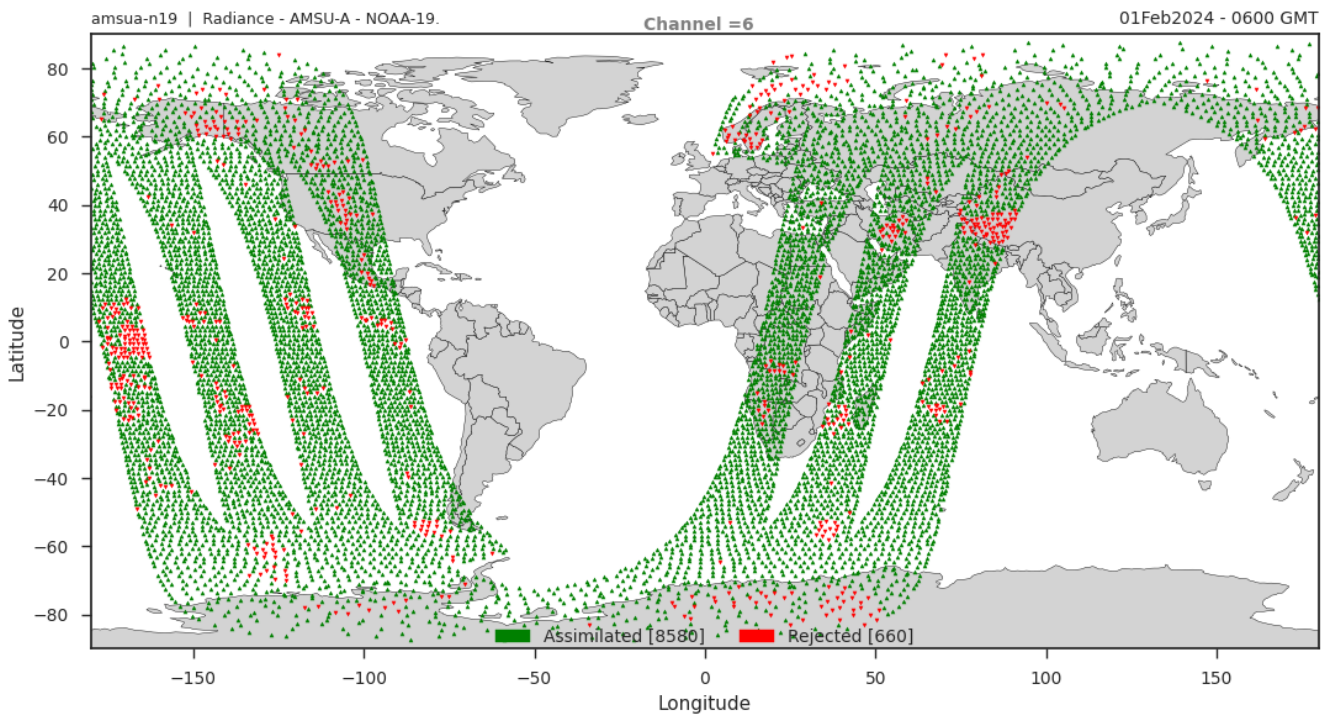
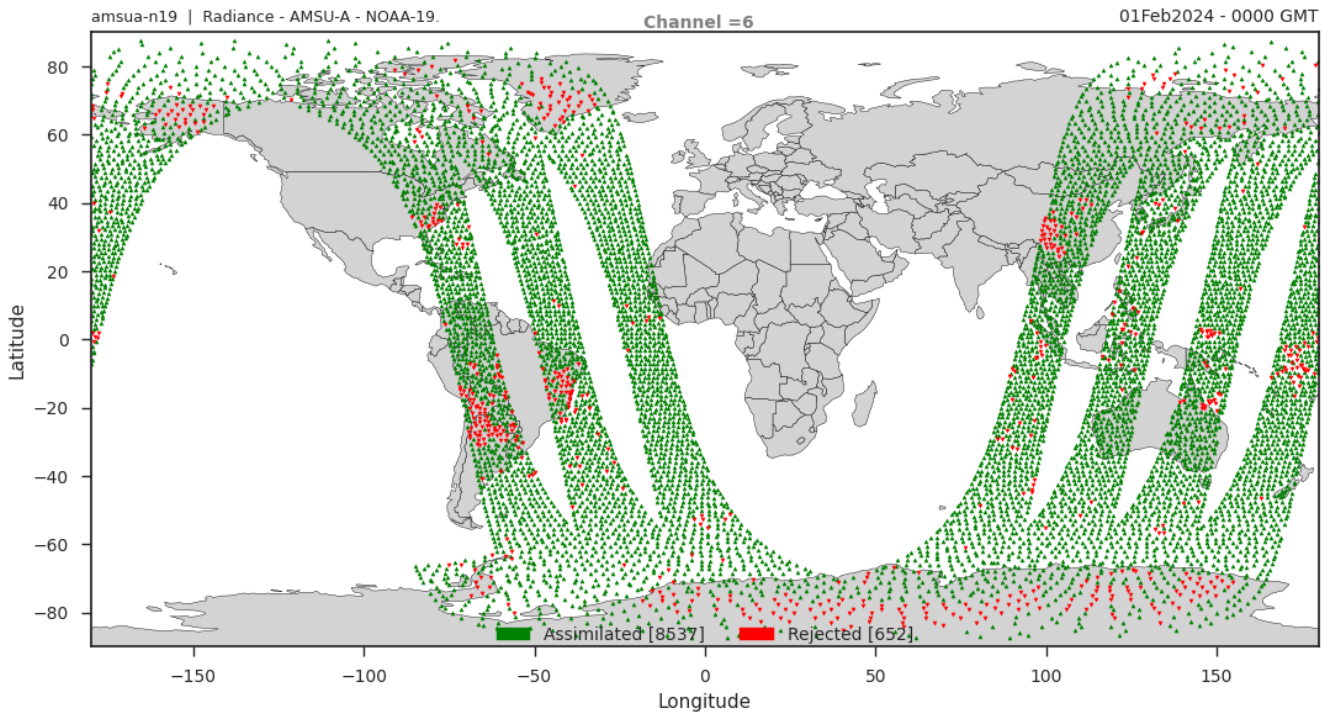
channel 6 not monitored on the date --> 2024-02-01:06

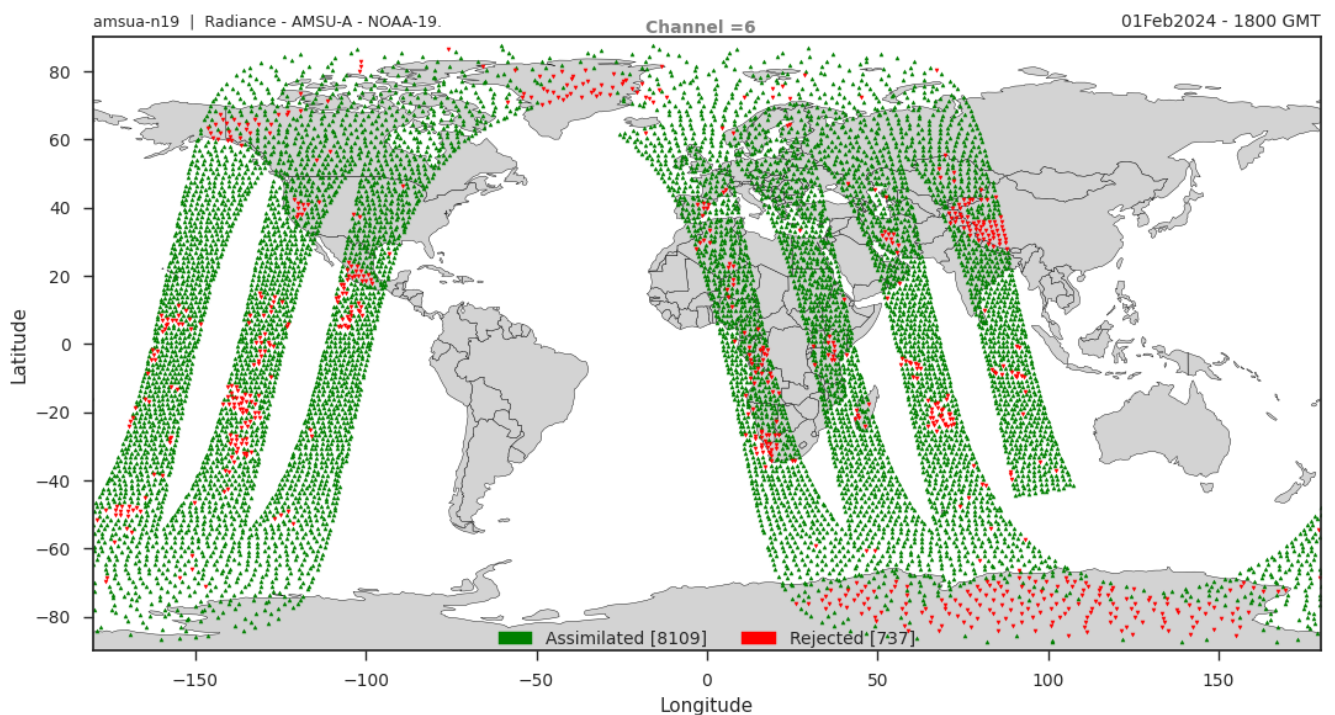
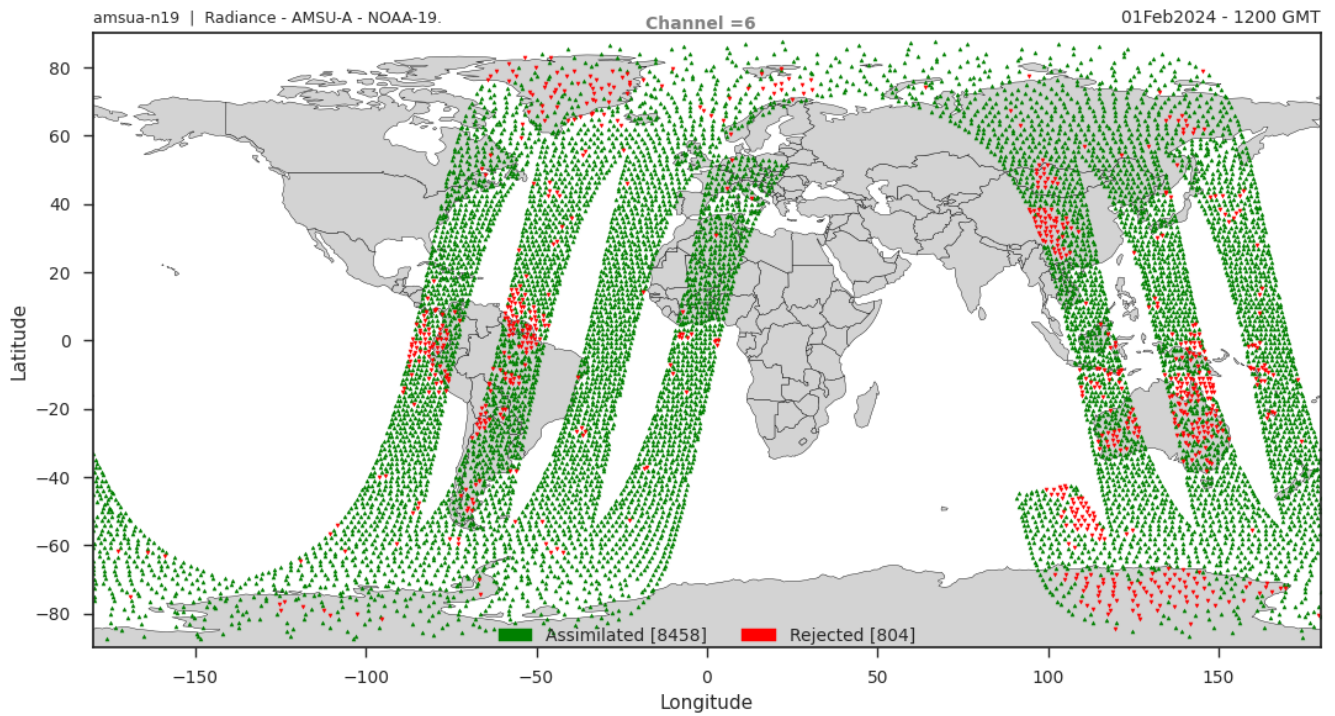
/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You
can get the original 'naturalearth_lowres' data from
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  path=gpd.datasets.get_path('naturalearth_lowres')
```

channel 6 not monitored on the date --> 2024-02-01:12

/home/radiancia/DiagRadi-V2_issue45/gsideag/__main__.py:93: FutureWarning: The
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You
can get the original 'naturalearth_lowres' data from
<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>.
path=gpd.datasets.get_path('naturalearth_lowres')

channel 6 not monitored on the date --> 2024-02-01:18





Observe que no exemplo acima os dados do canal selecionado foram assimilados ou rejeitados. Caso o canal também contenha algum dado monitorado, uma segunda figura seria gerada com a localização destes dados. Abaixo um exemplo de canal do satélite n19 cujo os dados são monitorados.

```
[24]: channel = 8
dateIni = '2024020100'
dateFin = '2024020118'

gd.plot_diag.statcount(gdf_list,
```

```
varName=varName,  
varType=varType,  
noiqc=False,  
dateIni=dateIni,  
dateFin=dateFin,  
nHour=nHour,  
channel=channel,  
figTS=False,  
figMap=True,  
markersize=4.80)
```

```
channel 8 not assimilated or rejected on the date --> 2024-02-01:00
```

```
/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The  
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You  
can get the original 'naturalearth_lowres' data from  
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.  
path=gpd.datasets.get_path('naturalearth_lowres')
```

```
channel 8 not assimilated or rejected on the date --> 2024-02-01:06
```

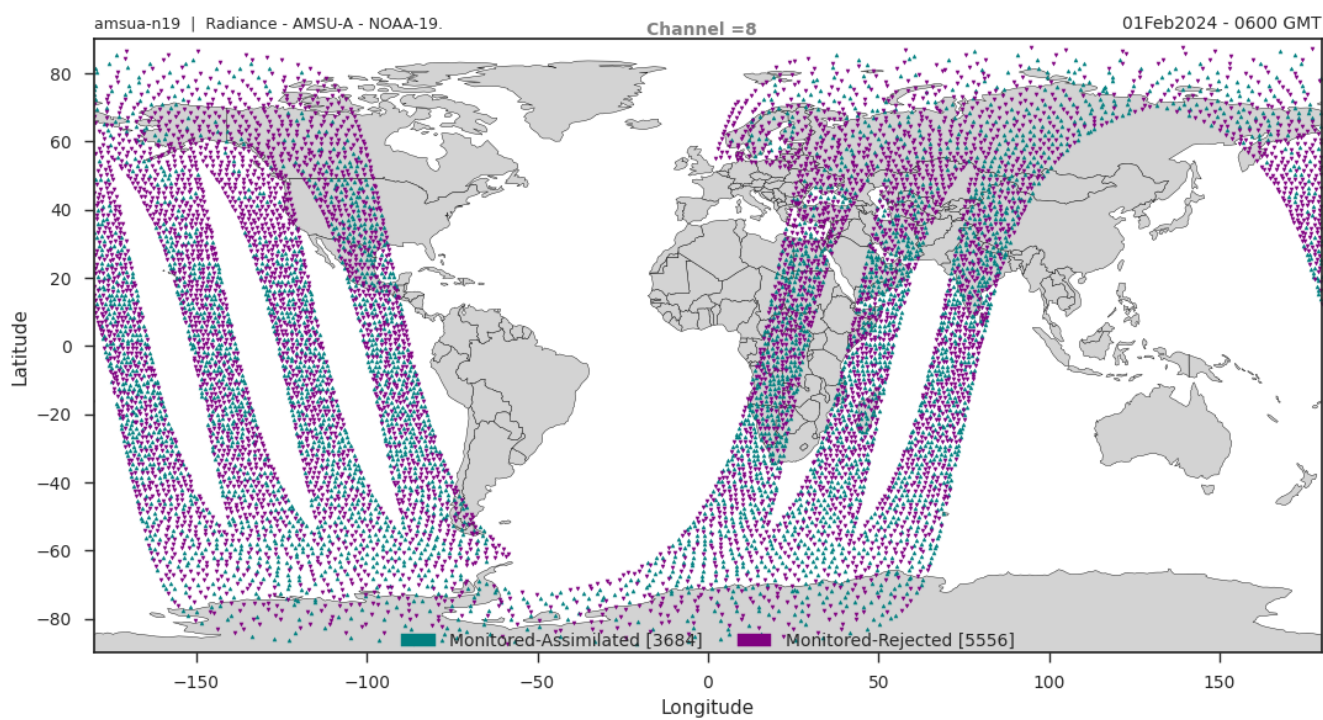
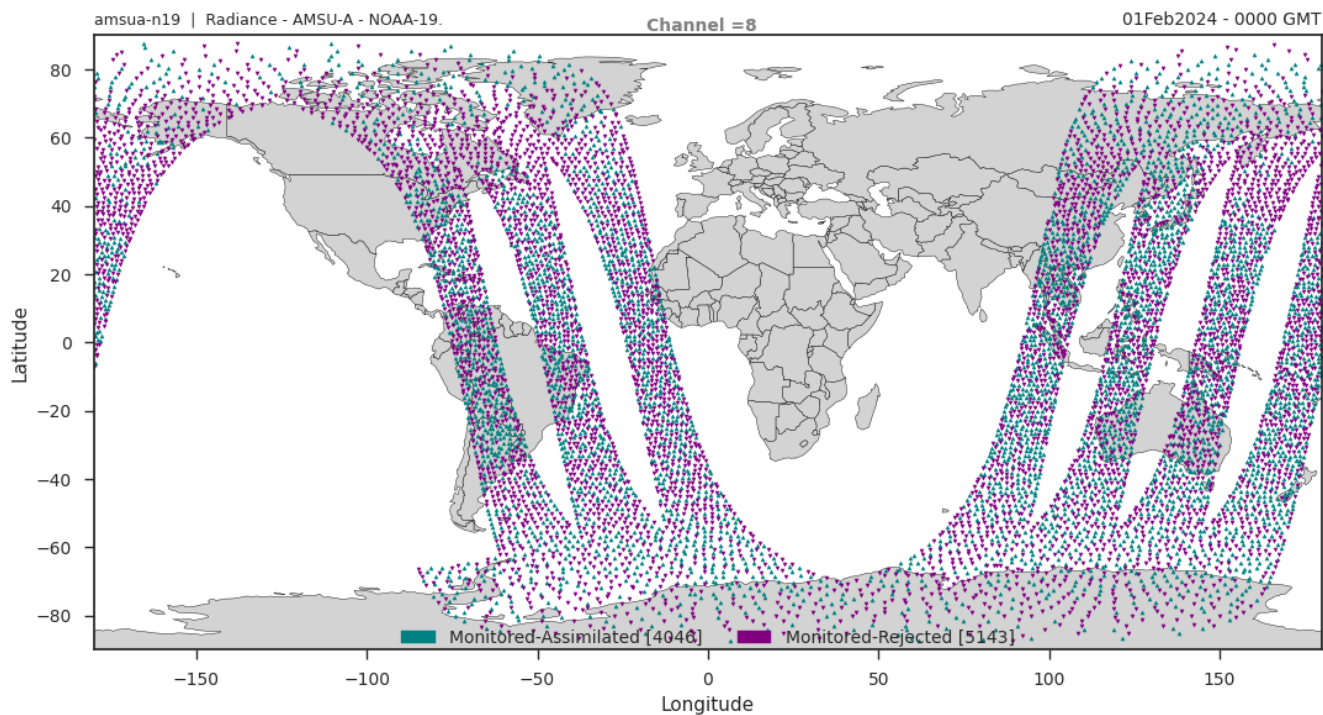
```
/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The  
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You  
can get the original 'naturalearth_lowres' data from  
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.  
path=gpd.datasets.get_path('naturalearth_lowres')
```

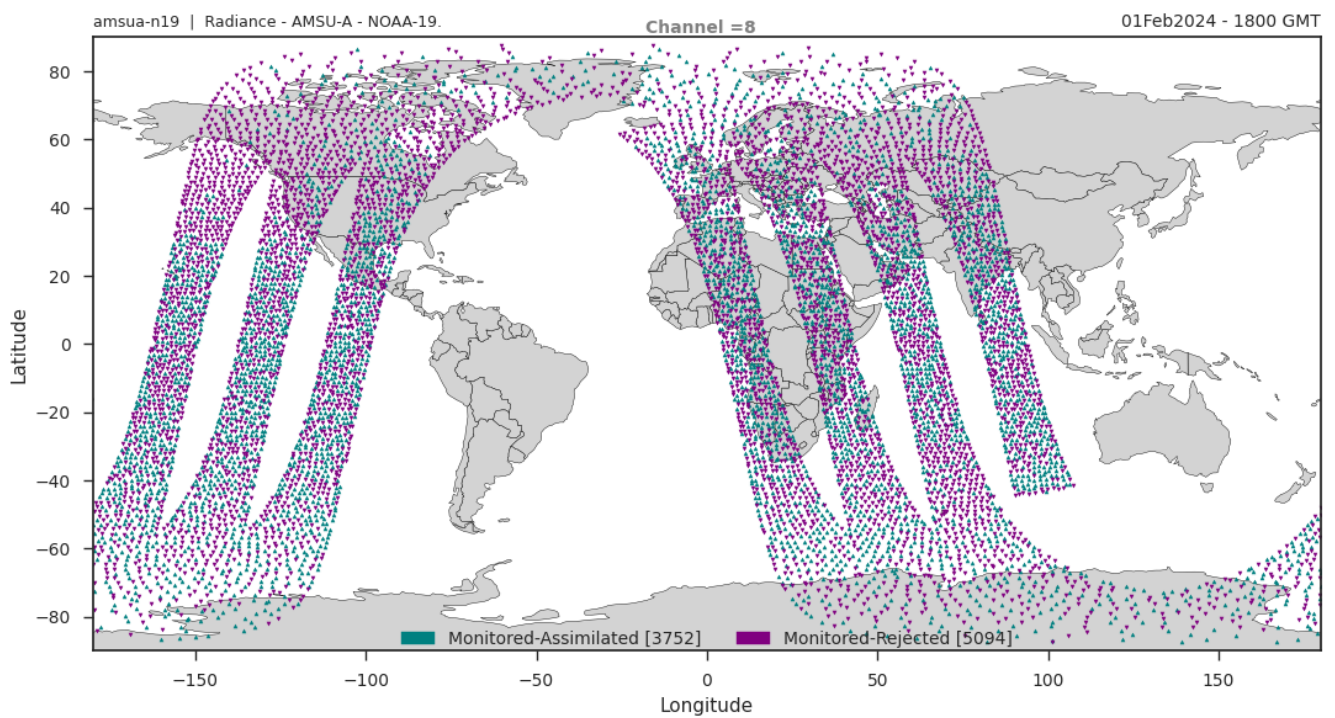
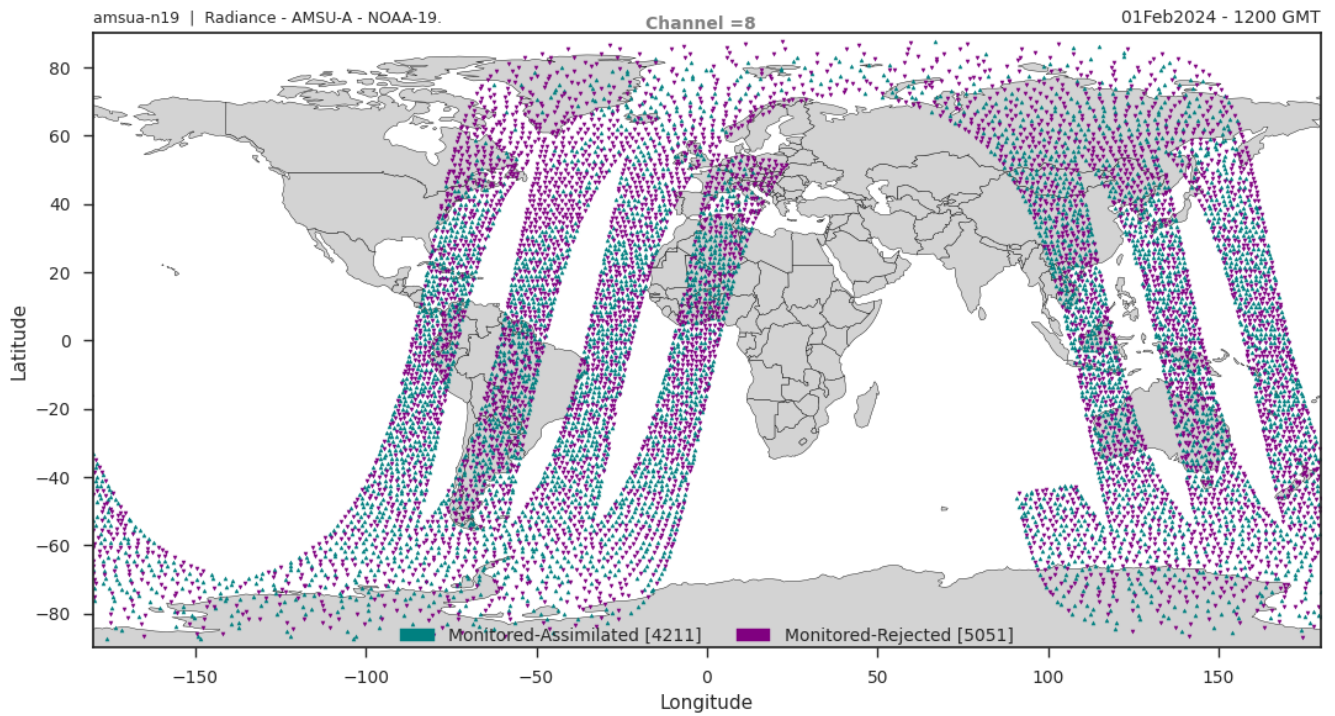
```
channel 8 not assimilated or rejected on the date --> 2024-02-01:12
```

```
/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The  
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You  
can get the original 'naturalearth_lowres' data from  
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.  
path=gpd.datasets.get_path('naturalearth_lowres')
```

```
channel 8 not assimilated or rejected on the date --> 2024-02-01:18
```

```
/home/radiancia/DiagRadi-V2_issue45/gsideiag/__main__.py:93: FutureWarning: The  
geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You  
can get the original 'naturalearth_lowres' data from  
https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.  
path=gpd.datasets.get_path('naturalearth_lowres')
```





Quando `figTS=True` apenas uma figura com a série temporal da quantidade de dados assimilados, rejeitados e monitorados é retornada.

```
[25]: channel = 6
dateIni = '2024020100'
dateFin = '2024020218'

gd.plot_diag.statcount(gdf_list,
                        varName=varName,
```

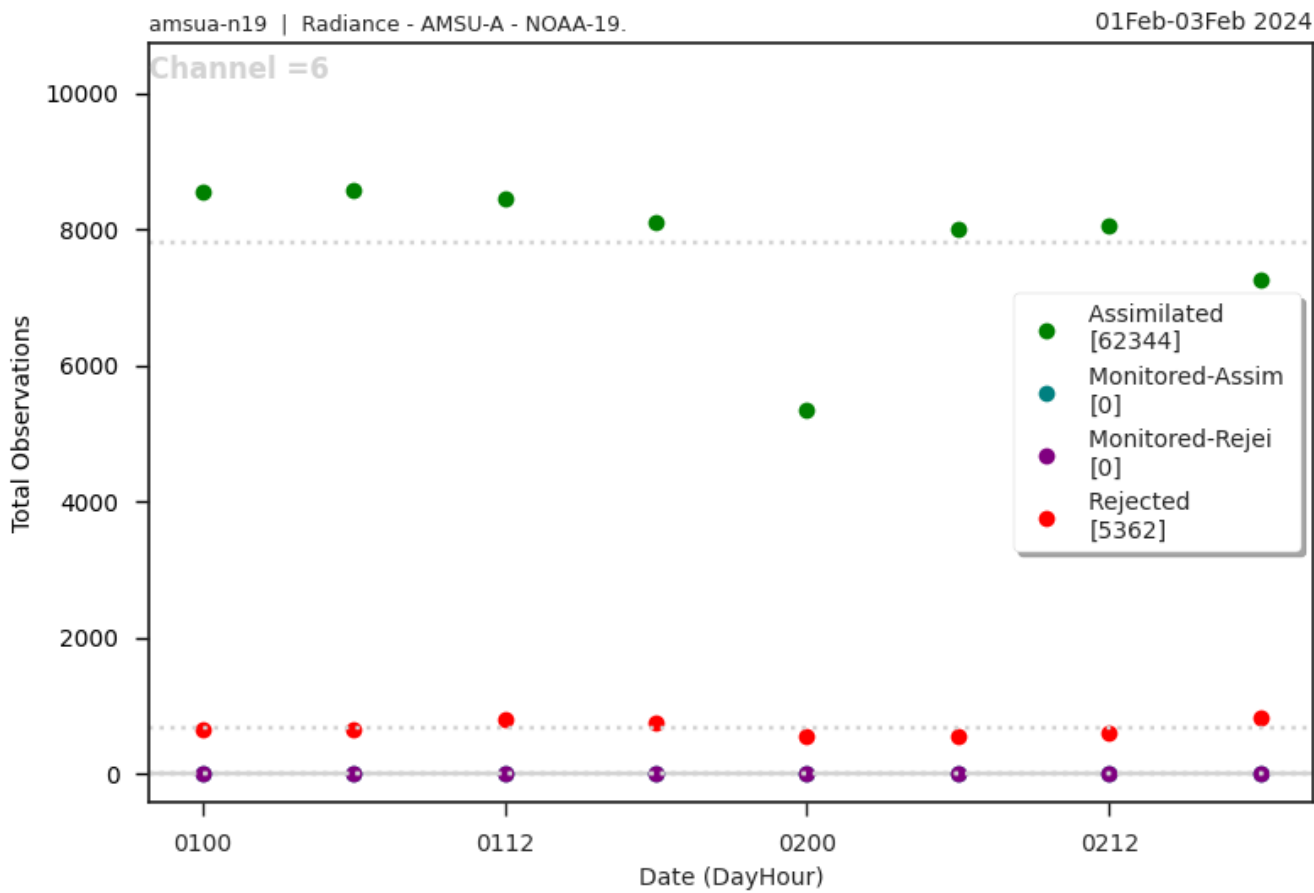
```

varType=varType,
noiqc=False,
dateIni=dateIni,
dateFin=dateFin,
nHour=nHour,
channel=channel,
figTS=True,
figMap=False,
markersize=4.80)

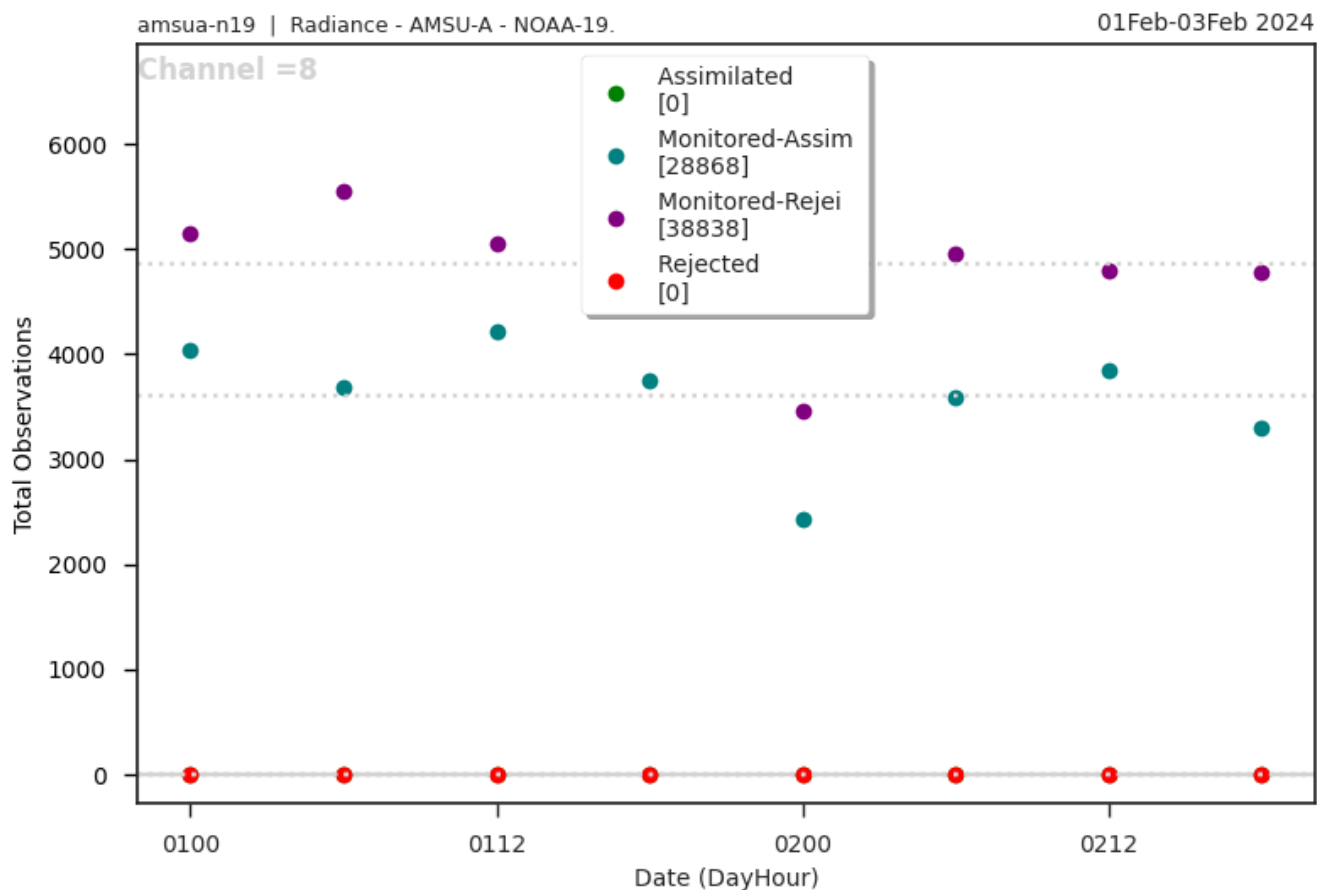
channel = 8
gd.plot_diag.statcount(gdf_list,
                        varName=varName,
                        varType=varType,
                        noiqc=False,
                        dateIni=dateIni,
                        dateFin=dateFin,
                        nHour=nHour,
                        channel=channel,
                        figTS=True,
                        figMap=False,
                        markersize=4.80)

```

<Figure size 600x400 with 0 Axes>



<Figure size 600x400 with 0 Axes>



Finalizado o uso dos arquivos, feche-os para liberar a memória utilizada:

```
[26]: for file in gdf_list:
      if file._FNumber != None:
          file.close()
```

O `readDiag` é um pacote em desenvolvimento e em constante atualização. Novas funcionalidades serão adicionadas e demonstradas por meio deste notebook.

```
[ ]:
```