HLS Snow Dynamics V1.1 Product User Guide 2018 -

Mitchell Bonney

Created: August 12, 2025

Last updated: January 7, 2026

Outline

Acronyms

- Fmask: Function of mask (v4.2)
- HLS: Harmonized Landsat Sentinel-2 (v2.0)
- IMS: Interactive Multisensor Snow and Ice Mapping System
- MGRS: Military Grid Reference System
- NASA: National Aeronautics and Space Administration
- NIR: Near infrared
- NOAA: National Oceanic and Atmospheric Administration
- NSIDC: National Snow and Ice Data Center
- STAC: Spatial Temporal Asset Catalog
- SWIR: Shortwave infrared

Terms

- *Certainty period*: Days with no observations between matching observations (i.e., snow/snow, non-snow/no-snow).
- *Chunk*: Single piece of dask-backed xarray data cube that can be accessed along with other chunks in parallel for faster processing.
- *Clear Observations* classified as land, water, snow, or high aerosol variants of those.
- *Constellation*: Satellites from the same program that are expected to be harmonized (i.e., Landsat 8 & 9, Sentinel-2A & 2B).
- *Implausible snow*: Range of dates (usually around summer) where it is implausible for a region to have any snow.
- *Inconsistent perennial*: Either start of end of winter year is snow covered.
- *Interannual*: Combined across all winter years.
- *Observation*: Single unique *clear* observation of Earth from HLS.
- *Perennial*: Every day in winter year is snow covered.
- *Pixel*: Small spatial unit (single location), represented as one x and y coordinate in data cube.
- *Scene*: Single HLS image with relevant metadata for initial filtering.
- *Snow cube*: Data cube (x, y, time) where each observation is snow, non-snow, or no data.
- *Snow free*: No day in winter year is snow covered.
- *Snow period*: Days in a row where snow is continuously observed.
- *Snow season*: Period from first to last snow observation in winter year.
- *Time-step*: Smallest temporal unit (single date), represented as one slice of time dimension in data cube.
- *Uncertainty period*: Days with no observations between mis-matched observations (i.e., snow/non-snow).
- *Unclear*: *Observations* classified as cloud adjacent, shadow, cloud, or fill.
- *Winter year*: Year-long period containing full snow season (i.e., snow minimum to next year snow minimum).

Open-source Python Modules

- cupy: GPU-accelerated numerical array computing
- dask: Parallel and distributed computing
- geopandas: Working with geospatial vectors
- hvplot: Interactive visualization
- jupyter: Web-based interactive development environment
- matplotlib: Visualization
- numpy: Numerical array computing
- pandas: Working with data frames
- polars: Working with data frames in parallel
- pyproj: Projections
- pystac_client: Working with STACs
- rasterio: Working with geospatial rasters
- rioxarray: Connecting rasterio and xarray
- shapely: Creating geospatial vectors
- tqdm: Progress bars
- stackstac: Turn STAC collections into xarray data cubes
- xarray: Working with multi-dimensional arrays
- xvec: Vector data cubes for xarray

Products – Snow Dynamics

- **start**: Start date (days since Dec 31) of the first (**F**) or biggest (**B**) snow period in winter year
  - **start_u**: Uncertainty (± days) of snow start (**F** or **B**)
  - *Interannual*: Weighted mean by uncertainty and implausibility (**_mn**) + uncertainty (**_u_mn**) and quality (**_q_mn**)
- **end**: End date (days since Dec 31) of the last (**L**) or biggest (**B**) snow period in winter year
  - **end_u**: Uncertainty (± days) of snow end (**L** or **B**)
  - *Interannual*: Weighted mean by uncertainty and implausibility (**_mn**) + uncertainty (**_u_mn**) and quality (**_q_mn**)
- **length**: Number of days with snow in total (**T**) or in the biggest (**B**) snow period during the winter year
  - **length_u**: Uncertainty (± days) of snow length (**T** or **B**)
  - *Interannual*: Weighted mean by uncertainty and implausibility (**_mn**) + uncertainty (**_u_mn**) and quality (**_q_mn**)
- **periods**: Number of separated snow periods during the winter year
  - Note: start (F/B), end (L/B) and length (T/B) don’t vary between versions unless periods > 1
  - *Interannual*: Mean (**_mn**)
- **status:** Snow status
  - 0: Seasonal (all other scenarios)
  - 1: Perennial (all days in snow year are snow)
  - 2: Inconsistent perennial (snow during peak summer – within week of snow year boundaries)
  - 3: Snow free (no days in snow year are snow)
  - 4: Ephemeral (longest snow period is a week or less)
  - *Interannual*: % years with perennial snow (**pPerennialSnow**), % years snow free (**pSnowFree**)

Other supported products

- Other interannual variants:
  - Mean, cleaned mean (>15 days uncertainty removed)
  - Best value (based on quality)
  - Standard deviations

V1.1 Product

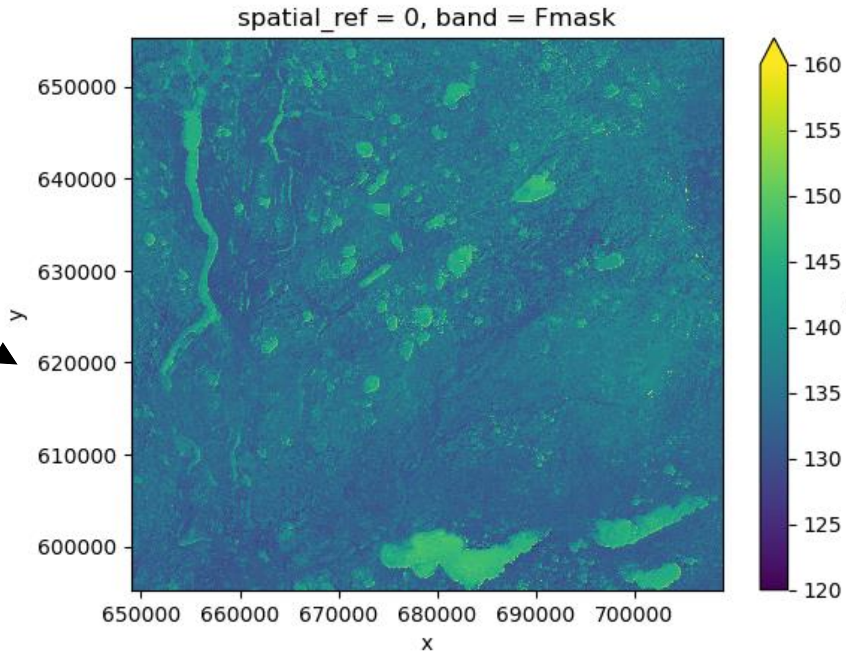
- Snow classifier: HLS Fmask v4.6
  - Binary: Snow vs. Non-snow for each observation
- Snow dynamics algorithm: Custom built system
  - Based on a peak finding algorithm on a cleaned binary time-series for each winter year

► Dimensions: (**winterYear**: 1, **x**: 2001, **y**: 2001)

► Coordinates: (4)

▼ Data variables:

snow_startF	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_startF_u	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_startB	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_startB_u	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_endL	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_endL_u	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_endB	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_endB_u	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_lengthT	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_lengthT_u	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_lengthB	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_lengthB_u	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_periods	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...
snow_status	(winterYear, y, x)	float32	dask.array<chunksize=(1, 501, 501), meta=np.ndarra...



Process

1. Study area (i.e., Canada) split into tiles (e.g., 60 x 60 km)
   - Steps 2-6 are completed for each tile in succession
2. All available HLS Fmask data accessed from STAC
3. Fmask data converted to binary snow cube
4. Snow cube cleaned to remove noise*
5. Two-year snow cube converted to winter year snow dynamics*
6. Winter year snow dynamics merged into interannual product*
   - Can also bring winter year products to final step
7. Once all tiles processed, merged into wide-area products

* = Tile-based statistics from IMS and climate datasets provide useful information for data cleaning and temporal filtering.

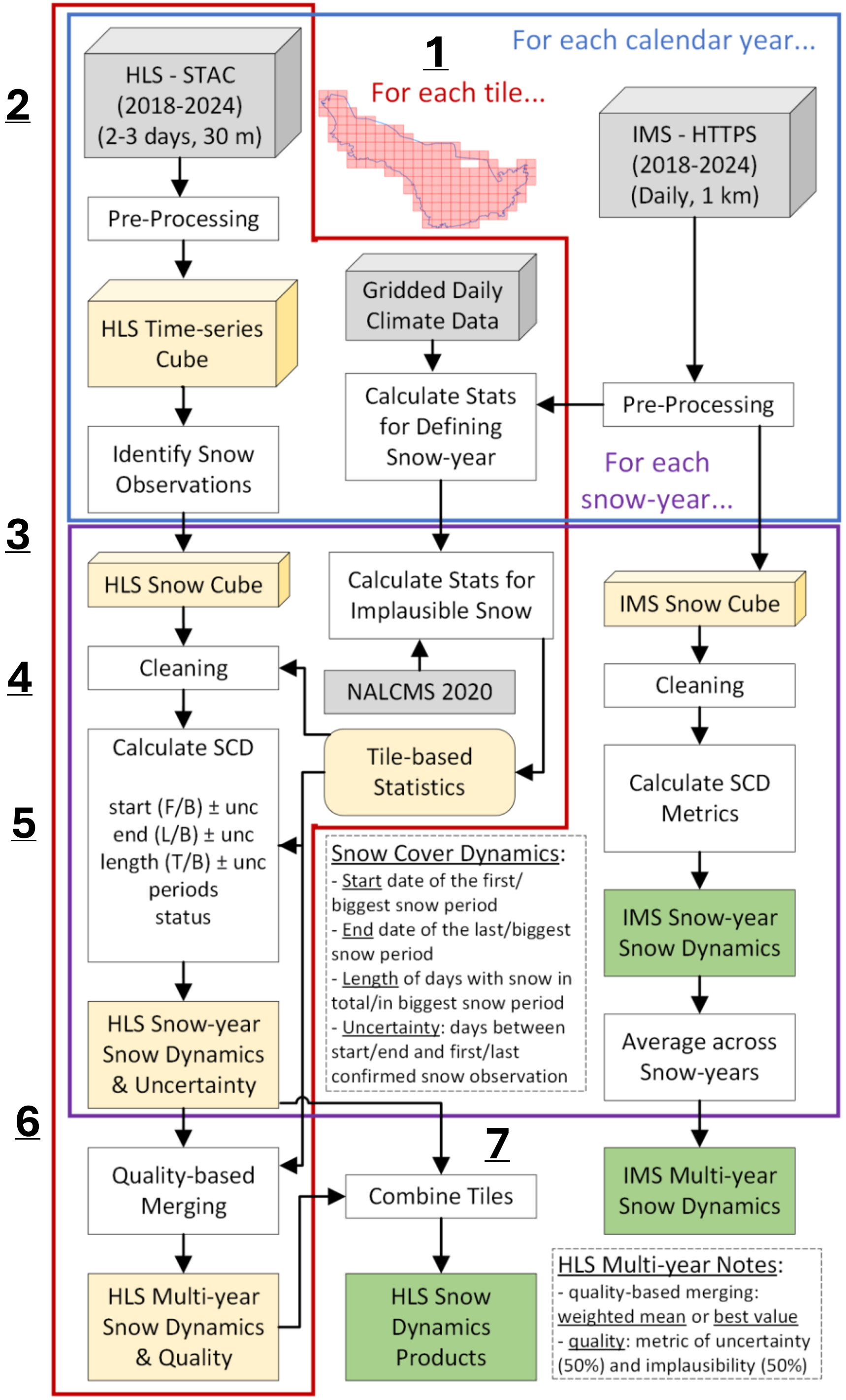
Full process is reproducible as modular Python API functions for easy use and documentation.

Overall Workflow

- Solid arrows: Complete
- Dashed arrows: In progress
- Dotted arrows: To do

Key Processing Tenets:

1. STAC + stackstac provide efficient and free access to entire HLS archive.
2. xarray + dask (LocalCluster) provide efficient parallelized processing of data cubes throughout workflow on single machine.
3. Tile-based processing keeps workload manageable with available memory, but algorithm primarily works at pixel level through time. Each pixel can be considered an independent output and tile-related artifacts are rare.

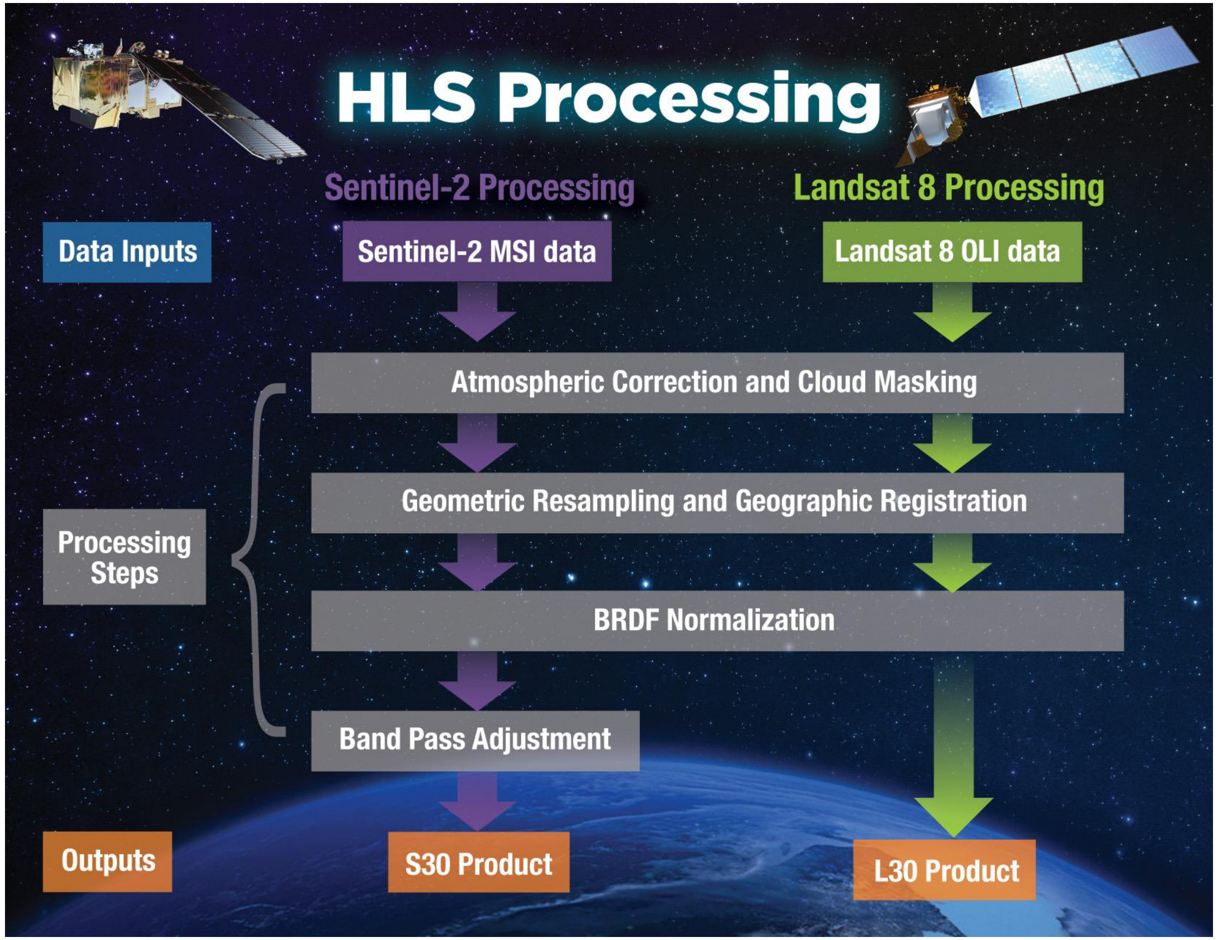


Calculating pixel-level IMS snow dynamics is not required.

Harmonized Landsat Sentinel-2 (1)

What is it?

- NASA-led effort to create an analysis-ready virtual constellation of Landsat 8+ and Sentinel-2 satellites
- Spatial resolution: 30 m
- Spectral resolution: blue, green, red, NIR, SWIR1, SWIR2
- Temporal resolution: varies from 16 days (2013) to ~2-3 days (2022)
- Global coverage

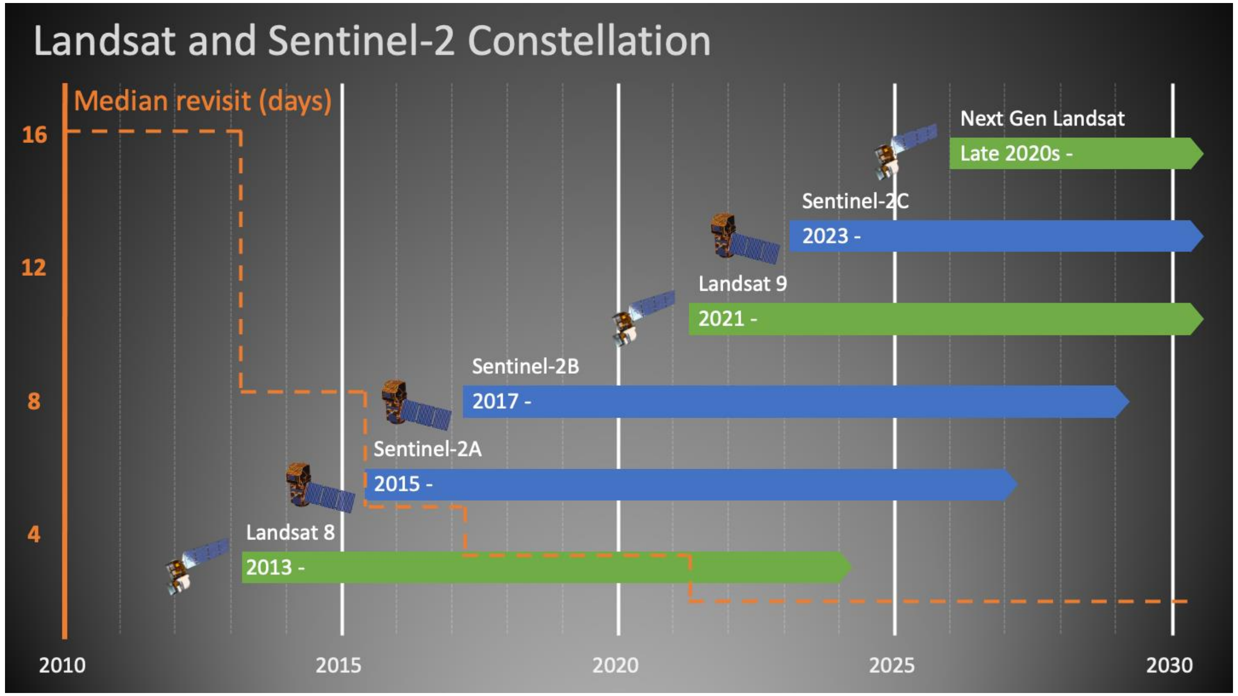


Harmonized Landsat and Sentinel-2 (HLS) | Earthdata (nasa.gov)

Harmonized Landsat Sentinel-2 (2)

Why is it useful?

- Only operational product that provides sufficient temporal resolution to consistently observe seasonal changes (e.g., snow dynamics, vegetation phenology) at landscape-scale
- Since beginning of Sentinel-2B collection in 2018, you can expect satellite overpasses every 2-3 days for most of the year globally
- Available for free through STAC, with improved (vs. Landsat C2) version of Fmask for cloud masking and V1 snow dynamics product
- Useful as primary data product to derive snow dynamics at 30 m

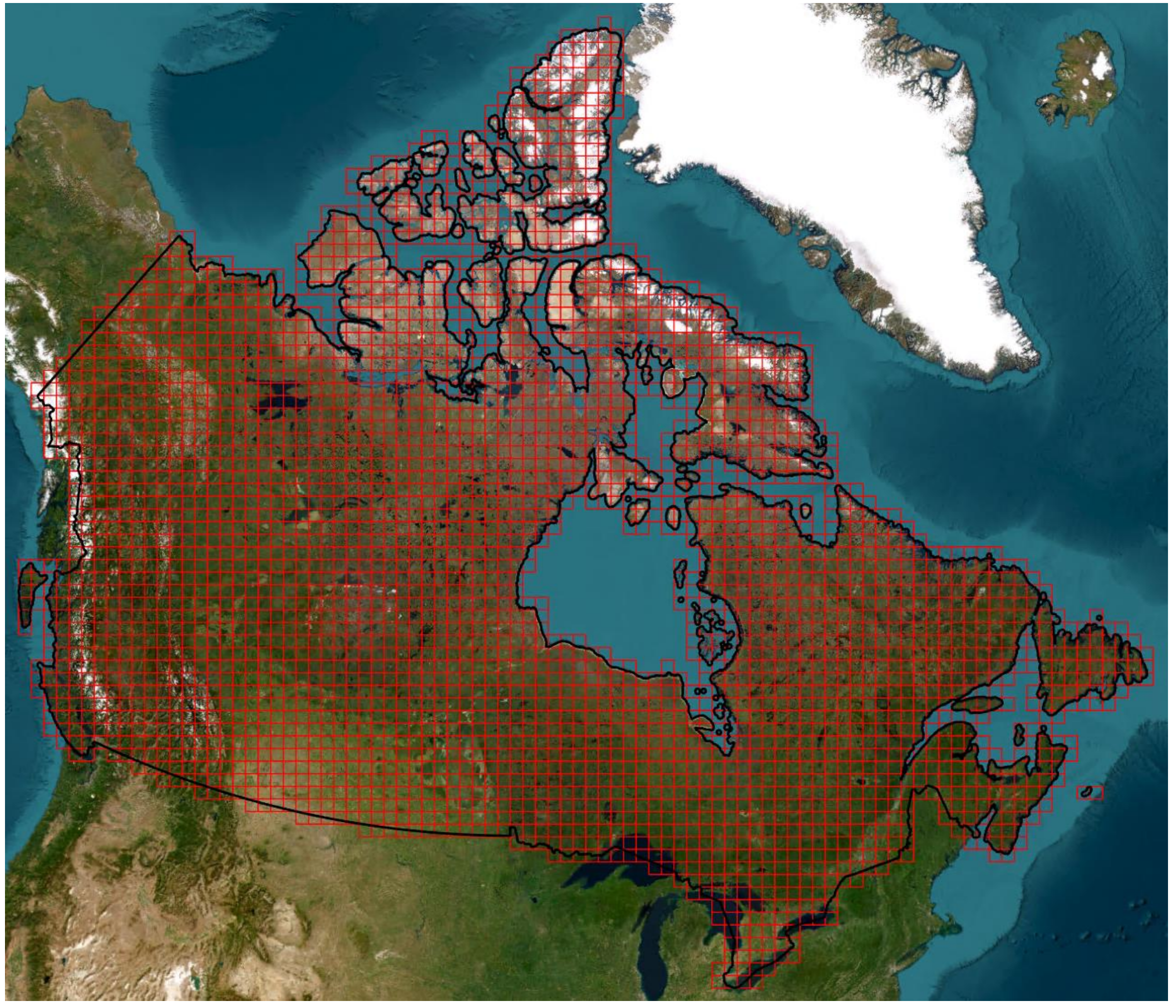


[Harmonized Landsat and Sentinel-2 (HLS) | Earthdata (nasa.gov)]

1. Area split into tiles

- Area: Canada + coastal waters (10 km)
- Size: 60 x 60 km
- Data in border tiles are clipped
- Canada: 3391 tiles

API functions:
- PreProcess_Utils.fishnet()



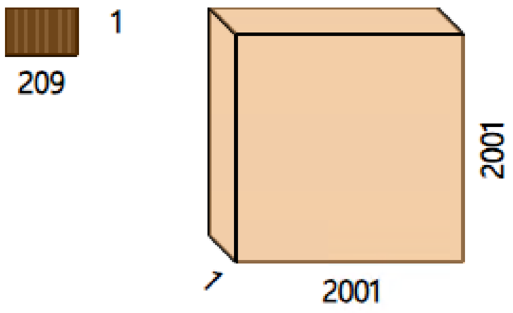
60 x 60 km tile fishnet over Canada.

2. Fmask accessed from STAC (1)

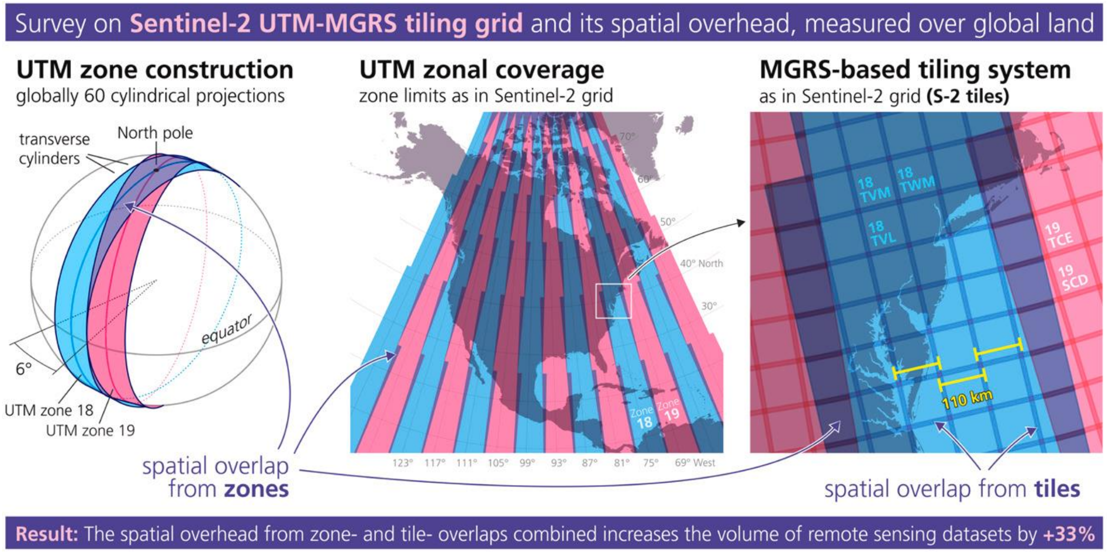
- For each tile and calendar year:
  - Find all HLS STAC items intersecting tile
  - Convert to dask-backed xarray cube
    - Fmask band, EPSG 3979, 30 m, nearest neighbor resampled, chunked by time-step, smallest data type
  - Remove bad scenes based on metadata
    - Missing times, cloud cover filter not applied here
  - Remove duplicate observations
    - Same-day observations from same constellation
    - First observation chosen

API functions:
- STAC_Utils.observationAvailabilityHLS()
  - STAC_Utils.accessSTAC()
  - PreProcess_Utils.poly2bbox()
  - PreProcess_Utils.removeBadScenes()
  - PreProcess_Utils.sameDayMerge()

	Array	Chunk
Bytes	798.07 MiB	3.82 MiB
Shape	(209, 1, 2001, 2001)	(1, 1, 2001, 2001)
Dask graph	209 chunks in 2 graph layers	
Data type	uint8 numpy.ndarray	



Yearly Fmask xarray cube for a tile with 209 time-steps and 2001x2001 pixels.

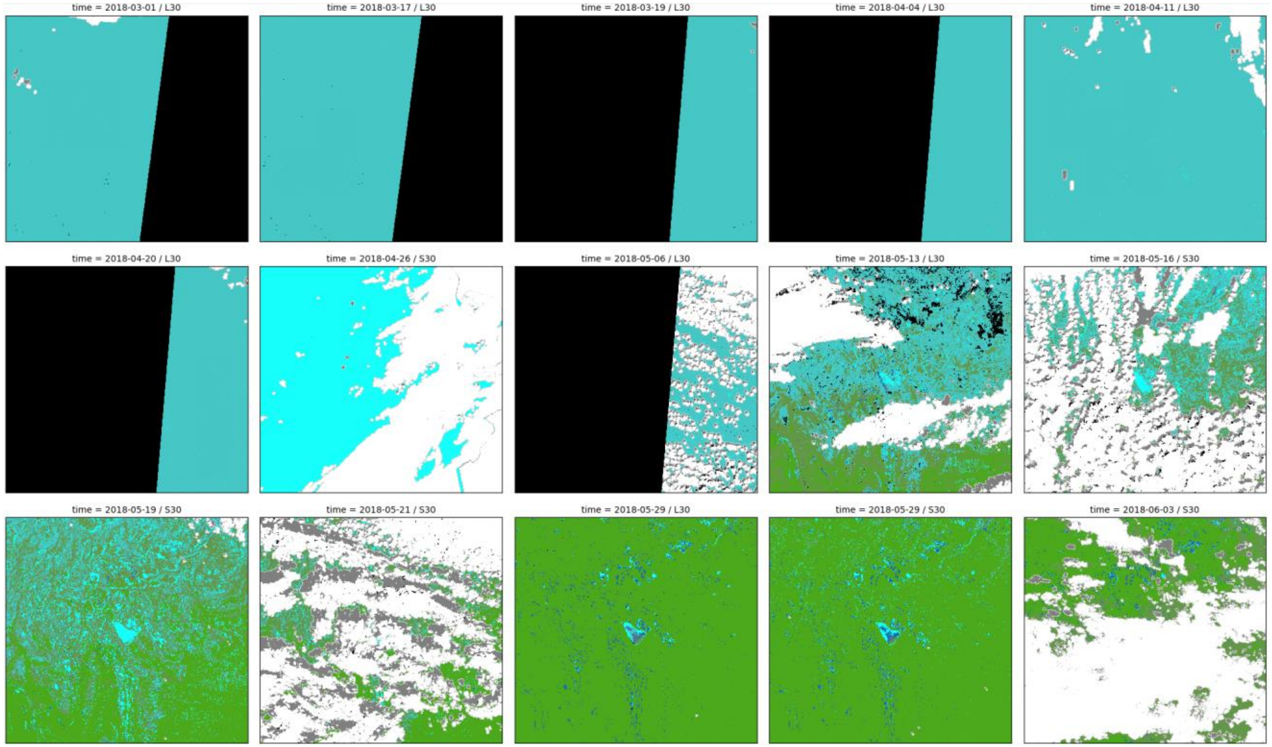


MGRS grid (used by HLS) leads to many duplicate observations (Bauer-Marschallinger and Falkner, 2023).

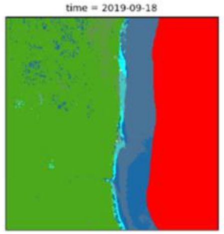
2. Fmask accessed from STAC (2)

- For each tile and calendar year:
  - Convert bit-packed Fmask values to categories based on hierarchy
    - fill > cloud > shadow > cloud adjacent > aerosol > snow > water > land
      - i.e., shadow covered water = shadow
  - For border tiles:
    - Mask Fmask values outside area of interest (i.e., Canada)
  - Download processed Fmask cube as NetCDF file
    - fmask_1238_2020.nc < Tile 1238, year 2020

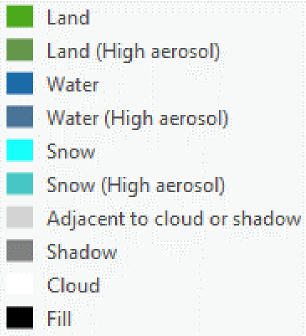
API functions:
- STAC_Utils.observationAvailabilityHLS()
  - PreProcess_Utils.convertFmask()
  - PreProcess_Utils.downloadNC()



Fmask observations during snow melt period in 2018 for a tile.



Masked border tile (red = outside area of interest).

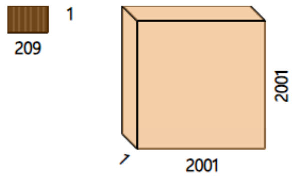


3. Fmask converted to snow cube

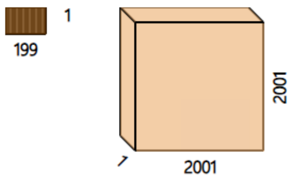
- For each tile and two-year winter year:
  - Load two calendar year Fmask cubes covering winter year (i.e., 2022 – 2023)
  - Make decision on same-day, different constellation, observations
    - Select lower value in hierarchy if different (i.e., land > cloud)
  - Concatenate into 2-year cube
  - Remove bad time-steps
    - > 99% of pixels are unclear
  - Reclassify into snow, non-snow, unclear
    - Snow = 1, non-snow = 0, unclear = NaN

API functions:
- PreProcess_Utils.uploadNC()
- SnowUtils.annualFmask2SnowCube()
  - PreProcess_Utils.timestepClean()

	Array	Chunk
Bytes	798.07 MiB	3.82 MiB
Shape	(209, 1, 2001, 2001)	(1, 1, 2001, 2001)
Dask graph	209 chunks in 2 graph layers	
Data type	uint8 numpy.ndarray	

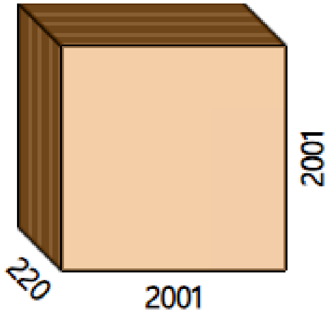


	Array	Chunk
Bytes	759.88 MiB	3.82 MiB
Shape	(199, 1, 2001, 2001)	(1, 1, 2001, 2001)
Dask graph	199 chunks in 2 graph layers	
Data type	uint8 numpy.ndarray	

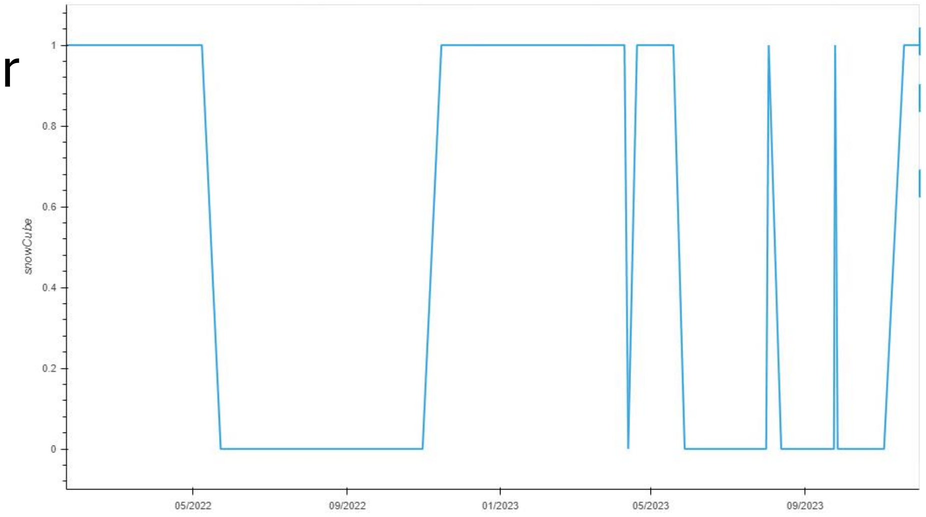


Two-year input Fmask cubes with 209 + 199 time-steps.

	Array	Chunk
Bytes	3.28 GiB	15.27 MiB
Shape	(220, 2001, 2001)	(1, 2001, 2001)
Dask graph	220 chunks in 699 graph layers	
Data type	float32 numpy.ndarray	



Two-year output snow cube with 220 time-steps.

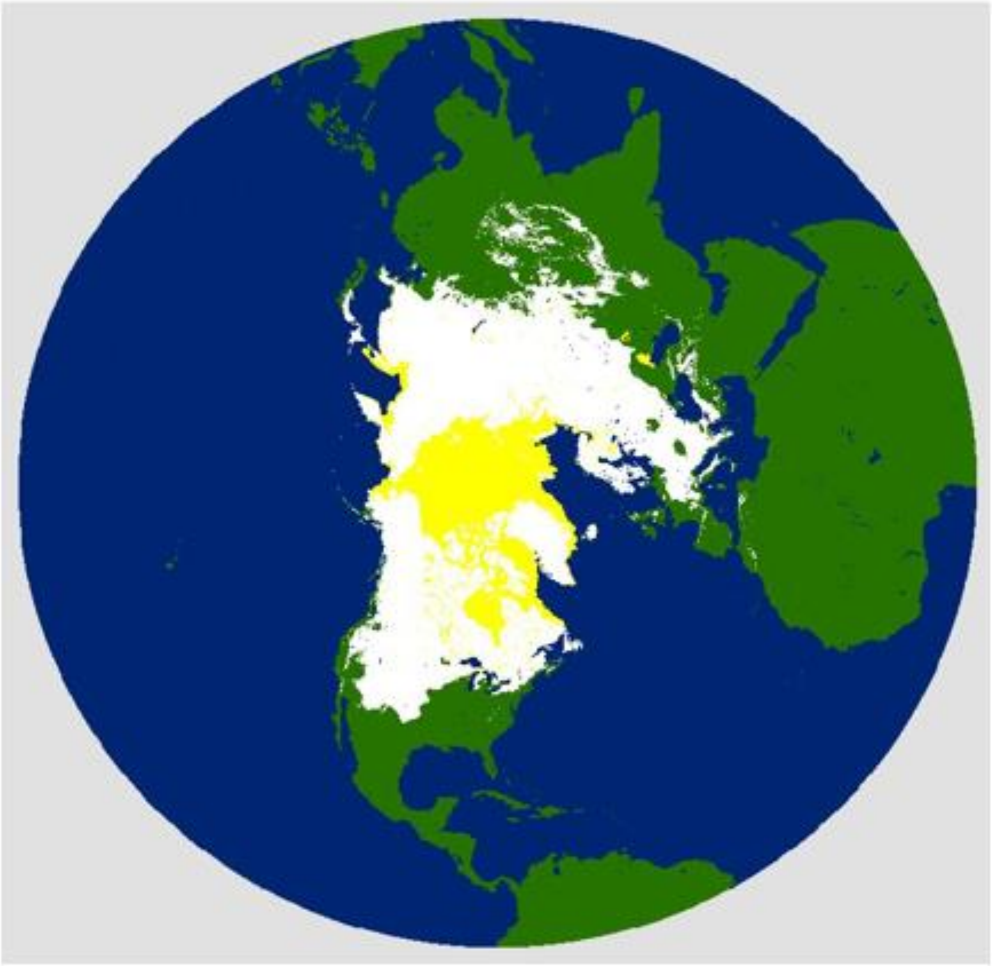


Single pixel time-series from two-year output snow cube, showcasing seasonal snow but with noise to clean. Gaps between observations filled in for clarity.

IMS 1 km Daily Snow and Ice (1)

What is it?

- 1 km daily product from NOAA NSIDC that classifies the northern hemisphere into land, water, snow, and ice
- Available in near-real-time since December 2014
- Built by analysts making informed decision on the class of each pixel from various sources (e.g., satellite, radar, models, ground stations)

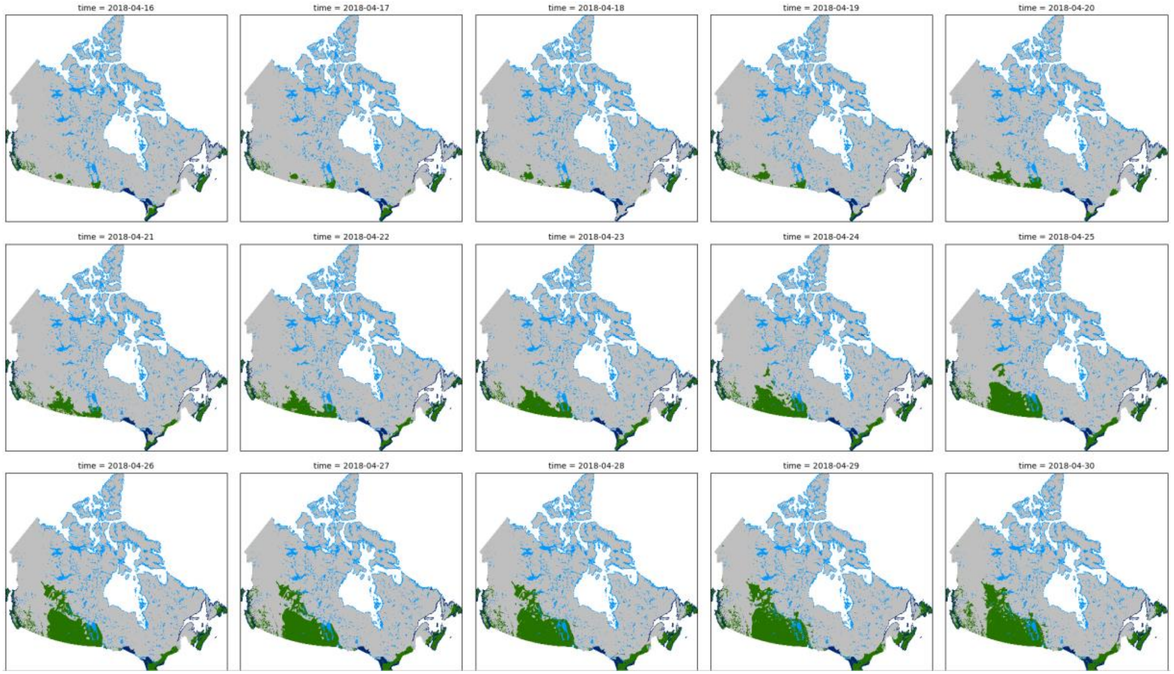


Single day IMS classification.

IMS 1 km Daily Snow and Ice (2)

Why is it useful?

- Since it is based on analyst interpretation, it can be considered a reference-level product of regional snow dynamics
  - Aligns (if sometimes delayed by a couple days due to clouds) with weather station data
  - Useful for cleaning and temporal filtering of HLS snow dynamics at tile level

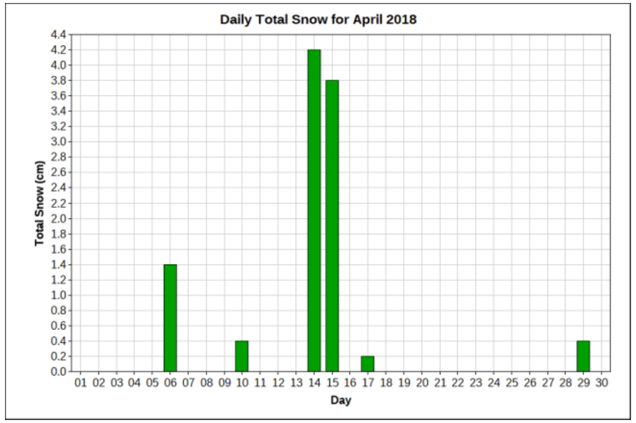


Canada-wide IMS classification from April 16 to April 30, 2018.

Apr 7

```
array([[    0.,     0.,     0.,     0.,     0.,     0., 1422.,   791.,     0.,
            0.,     0.,     0.,     0.,     0.,     0.,     0.,     0., 1419.,
         1419.,   725.,   102.,    20.,     0.,     0.,     0.,     0.,     0.,
            0.,     0.,     0.]])
```

Apr 18



Number of snow pixels over Toronto tile vs. Pearson Airport weather station snow fall for April 2018.

Tile-based Statistics

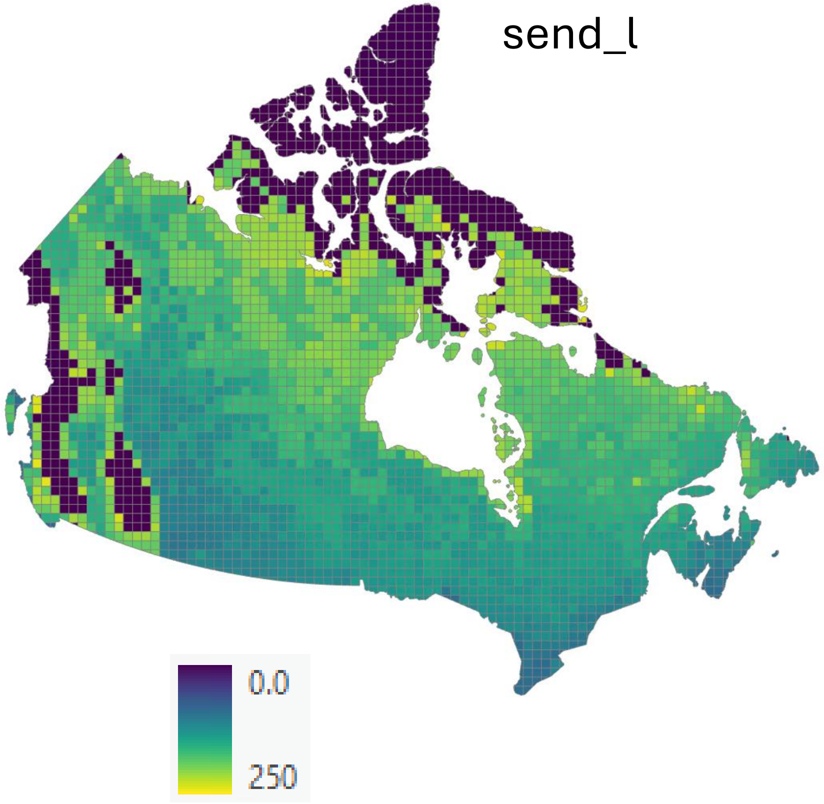
- Statistics for defining implausible snow period:
  - send_l: Latest winter year snow end date from IMS (2018-2024)
    - Must be at least 90 (i.e., March 31)
  - sstart_f: Earliest winter year snow start date from IMS (2018-2024)
    - Cannot be later than 335 (i.e., December 1)
  - sper_p: % coverage of tile by NALCMS 2020 perennial snow
- Statistics for defining winter year boundary
  - avg_yr: Average of smin_yr, tmax_yr, rmax_yr in days
    - smin_yr: Date with minimum snow cover from IMS
    - tmax_yr: Date with the maximum average temperature from gridded climate data
    - rmax_yr: Date with maximum solar radiation from gridded climate data
    - All variables 31-day smoothed (median: smin, mean: tmax, rmax)
- All variables in days since December 31

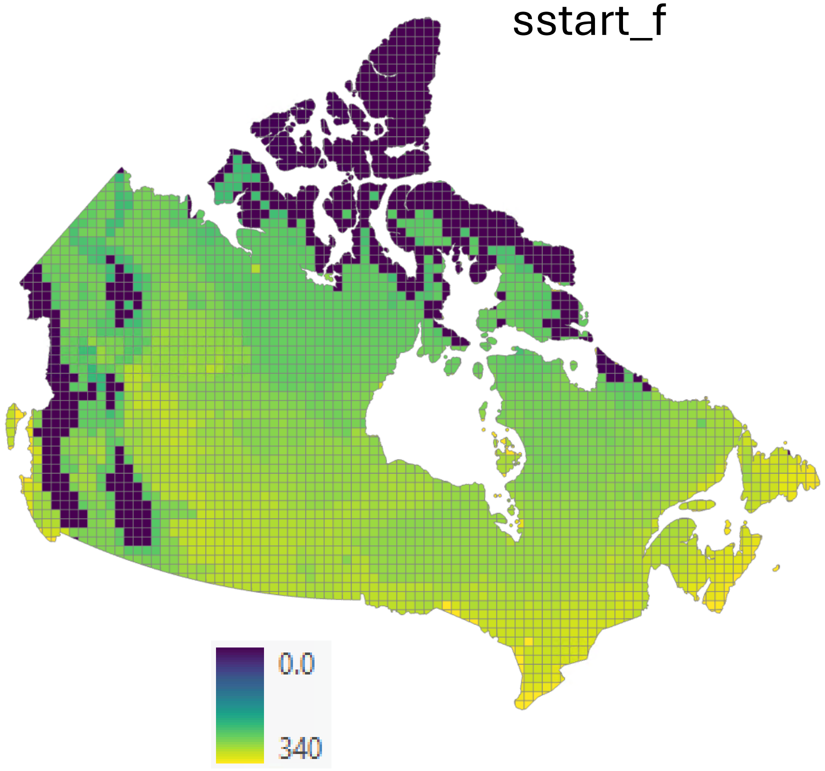
Assuming it is always possible for there to be snow between December and March anywhere in Canada.

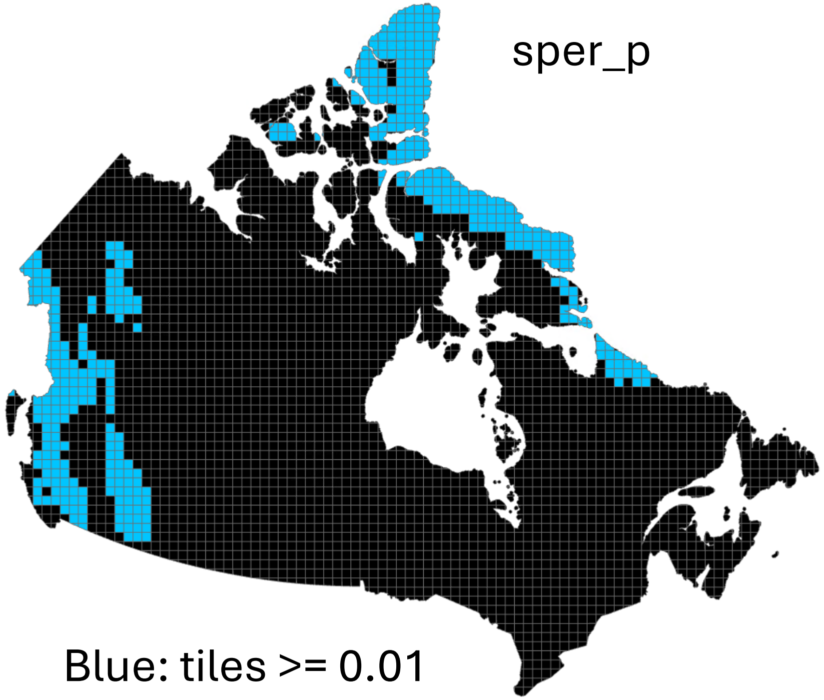
	send_l	sstart_f	sper_p	avg_18	avg_19	avg_20	avg_21	avg_22	avg_23	avg_24
0	90.0	335.0	0.000000	210.0	199.0	189.0	200.0	197.0	181.0	178.0
1	90.0	335.0	0.000000	210.0	201.0	199.0	201.0	200.0	190.0	187.0
2	90.0	335.0	0.000000	209.0	200.0	192.0	205.0	197.0	190.0	187.0
3	90.0	335.0	0.000000	208.0	201.0	186.0	190.0	199.0	179.0	182.0
4	90.0	335.0	0.000000	210.0	198.0	188.0	200.0	197.0	181.0	178.0
...	...	...	...	...	...	...	...	...	...	...
3386	NaN	NaN	20.354541	197.0	199.0	199.0	194.0	201.0	196.0	193.0
3387	NaN	NaN	11.178335	200.0	190.0	198.0	194.0	201.0	196.0	195.0
3388	NaN	NaN	7.835338	198.0	202.0	198.0	193.0	198.0	184.0	193.0
3389	NaN	NaN	0.041697	196.0	200.0	192.0	204.0	191.0	194.0	194.0
3390	NaN	NaN	0.000000	178.0	179.0	184.0	174.0	175.0	176.0	180.0

Tile-based statistics for first (most southern) and last (most northern) few tiles.

Tile-based Statistics (implausible snow)



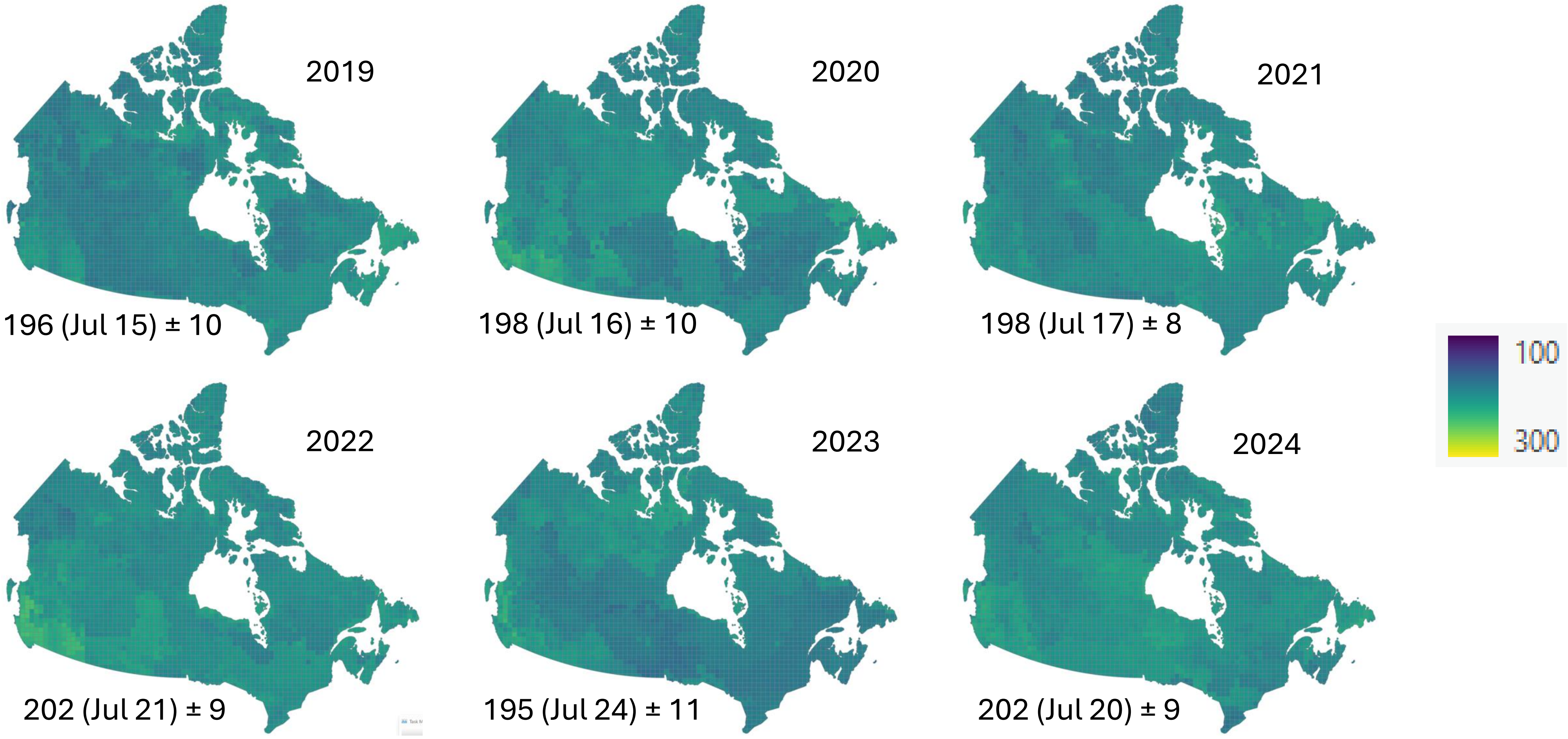




Blue: tiles >= 0.01 % perennial snow

Purple: Perennial tiles with no send_l or sstart_f

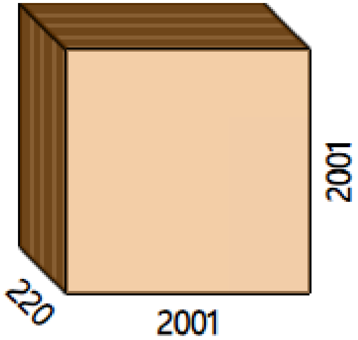
Tile-based Statistics (avg_yr, last 6 years)



4. Snow cube cleaned

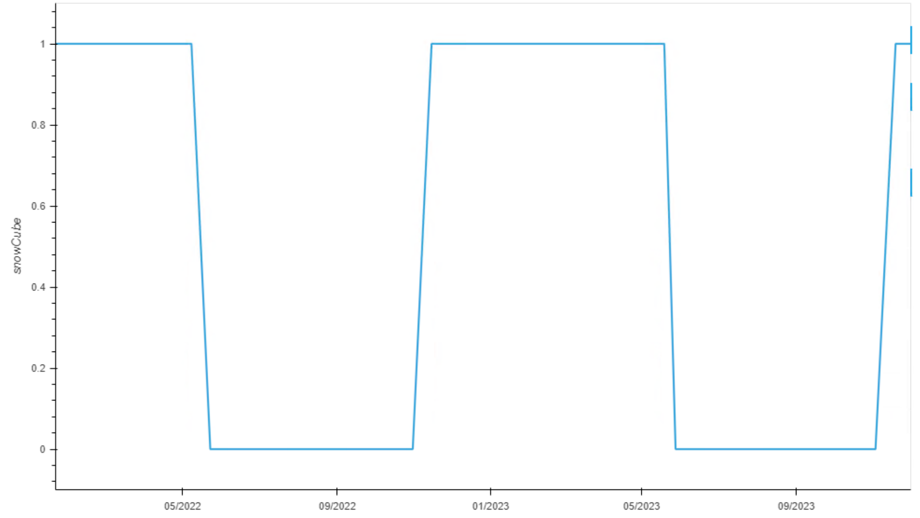
- For each tile and two-year winter year:
  - Change all observations during implausible snow period to non-snow
    - Implausible snow defined by IMS tile-based snow statistics
      - Week after send_l to week before sstart_f
      - No implausible snow if sper_p >= 0.01
  - Keep only periods when 2 (threshold) consecutive snow observations occur
    - For each pixel time-series, calculate cumulative sum that resets when non-snow is found
    - Resetting cumulative sum must reach 2 or observations are removed (NaN)
  - Do the same as above, but for consecutive non-snow observations
    - Removing isolated snow observations

	Array	Chunk
Bytes	3.28 GiB	15.27 MiB
Shape	(220, 2001, 2001)	(1, 2001, 2001)
Dask graph	220 chunks in 699 graph layers	
Data type	float32 numpy.ndarray	



Two-year cleaned snow cube with 220 time-steps.

[163, 272] < Implausible snow dates for tile (904): Jun 12 to Sep 29



Same pixel time-series from step 3, cleaned to remove implausible or isolated observations. Gaps between observations filled in for clarity.

API functions:
- Snow_Utils.cleanSnowCube()

5. Snow cube to snow dynamics (1)

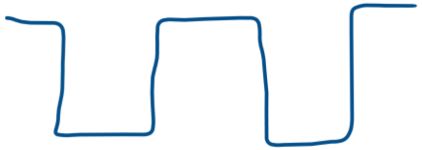
```
Snow status: 0
Snow periods: 1
Snow start: -53.0
Snow start uncertainty: 7.0
Snow end: 143.0
Snow end uncertainty: 4.0
Snow length: 197
Snow length uncertainty: 11.0
```

- For each tile and two-year winter year:
  - For each pixel time-series (Jan 1, yr1 to Dec 31, yr2):
    - Remove unclear observations
    - Use a peak finder algorithm (scipy's find_peaks) to generate information on peaks (i.e., snow periods)
      - For each peak where its middle point is within the winter year, as defined by avg_yr:
        - Calculate snow dynamics variables (i.e., peak length, start, end, uncertainties based on days until previous or next non-snow observation)
        - Add 1 to periods count
        - For biggest peak, calculate B variants of length, start and end and their uncertainties
      - If no peaks were found, confirm edge-case scenarios and apply calculations as needed:
        - Perennial snow (all observations = 1); Snow free (all observations = 0)
        - Snow-to-snow, but non-snow in between:
        - Non-snow to snow; or Snow to non-snow or
    - Confirm status (perennial, inconsistent [snow during peak summer], snow free, ephemeral [longest snow period is a week or less])
  - Create and save snow dynamics products (see following slides for details)

API functions:
- Snow_Utils.snowCube2SnowDynamics()
  - Snow_Utils.snowDynamics1D()
- PreProcess_Utils.downloadNC()

Most common 2-year time-series:

1 snow period

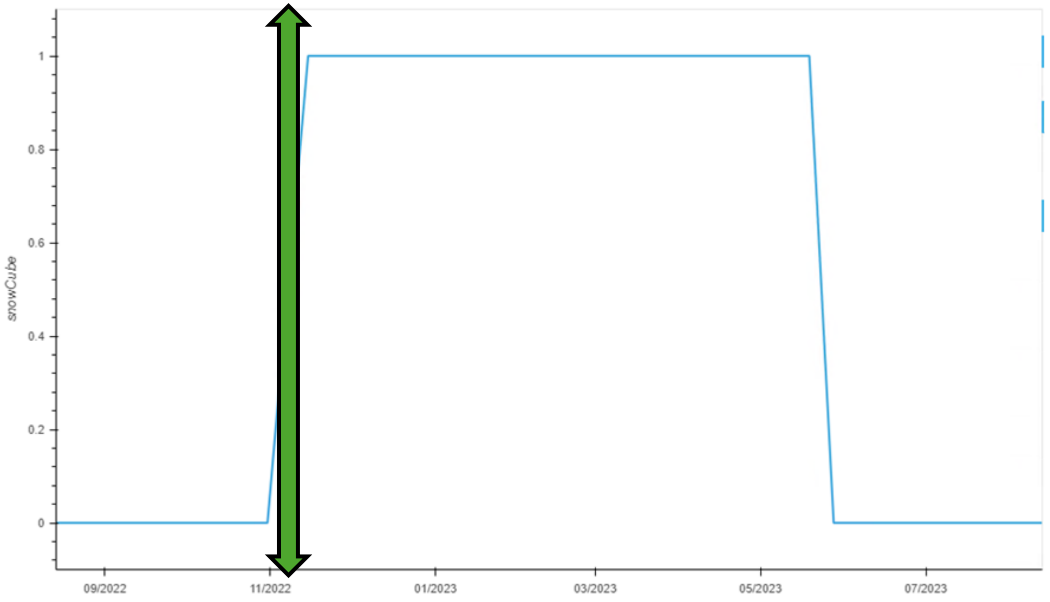


2 snow periods:

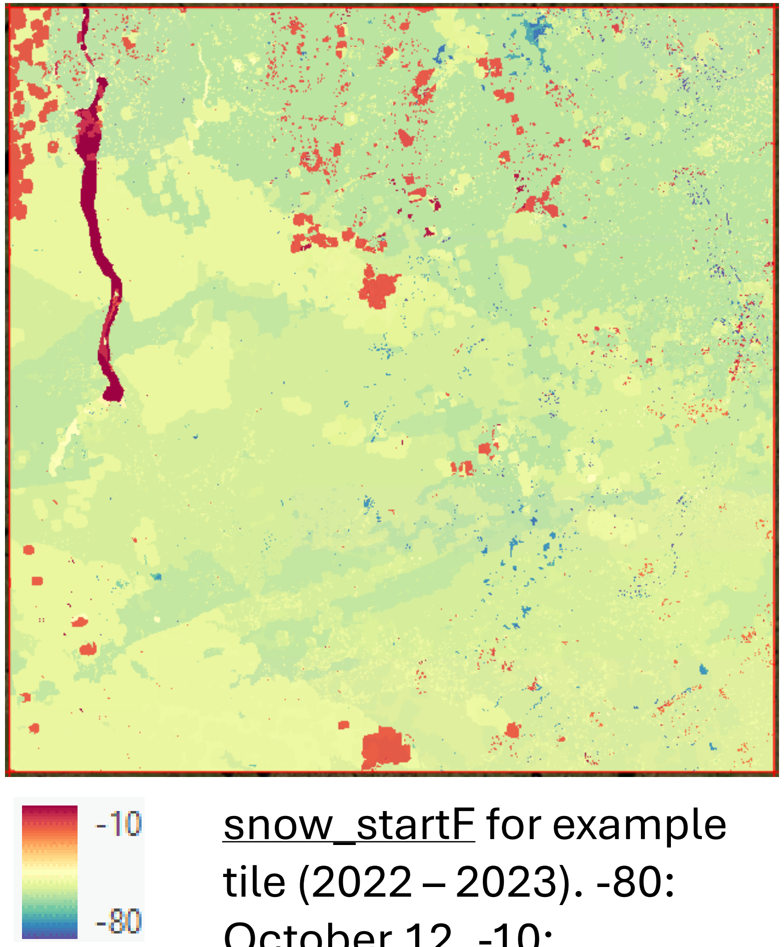
(End (B) will differ from End (L), periods = 2)

6. Snow cube to snow dynamics (2)

- <u>snow_start</u>
  - Definition: Start date of first (F) or biggest (B) snow period in winter year
    - Number of days from December 31 in winter year (Dec 31 = 0)
  - Pixel-level Calculation:
    - Middle date (rounded up) between left base of peak (previous non-snow) and next (first snow) observation



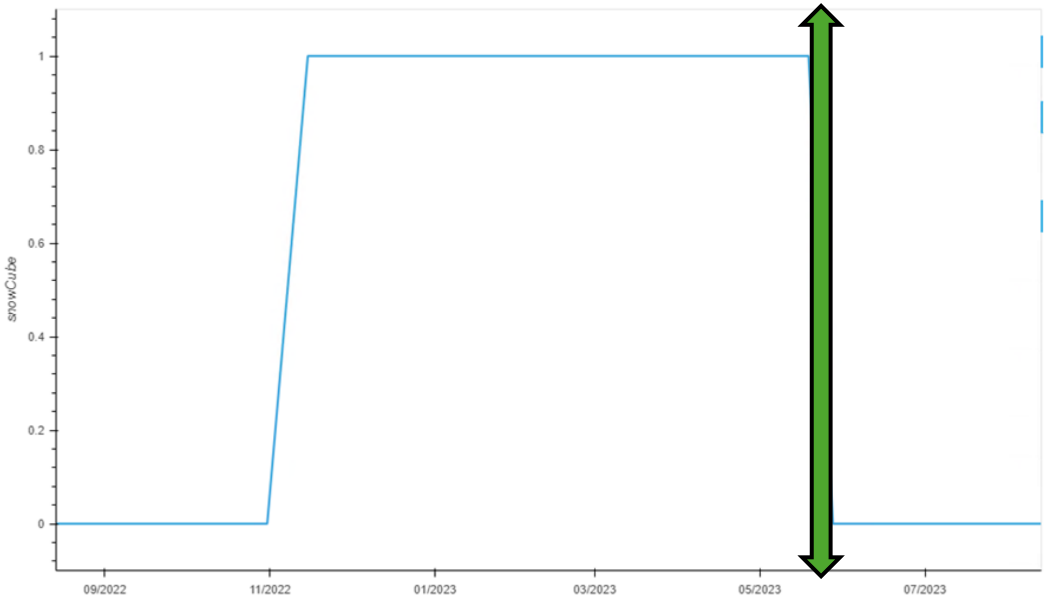
<u>snow_start</u> example for single pixel time-series from snow cube. <u>snow_start</u> will occur halfway along uncertainty period.



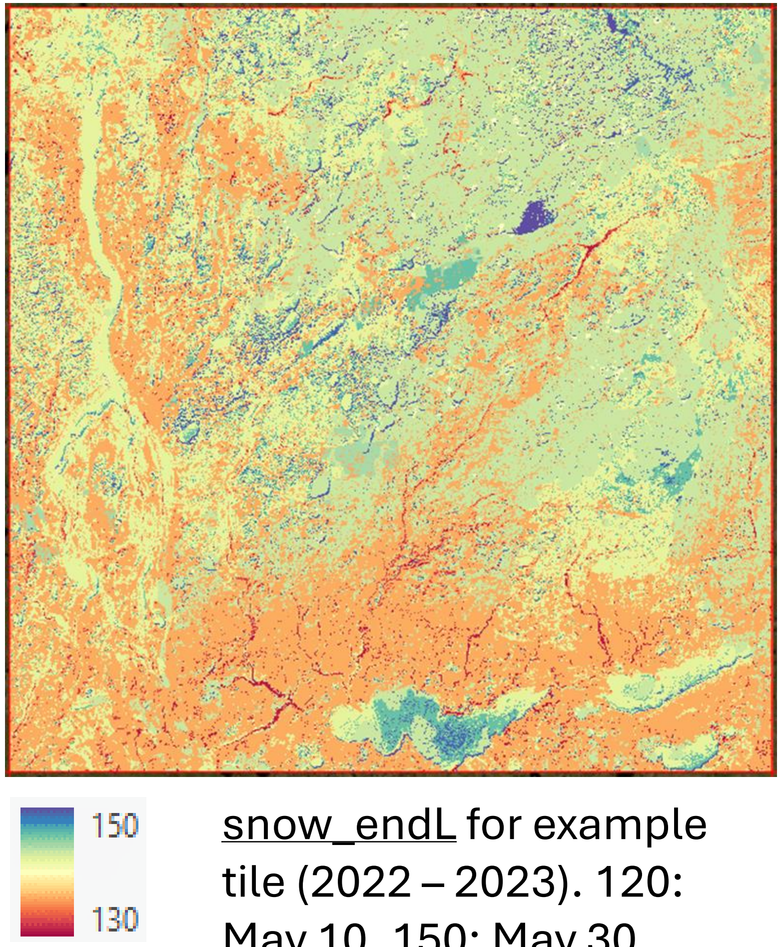
<u>snow_startF</u> for example tile (2022 – 2023). -80: October 12, -10: December 21.

6. Snow cube to snow dynamics (3)

- snow_end
  - Definition: End date of last (L) or biggest (B) snow period in winter year
    - Number of days from December 31 in winter year (Dec 31 = 0)
  - Pixel-level Calculation:
    - Middle date (rounded down) between right base of peak (next non-snow) and previous (last snow) observation



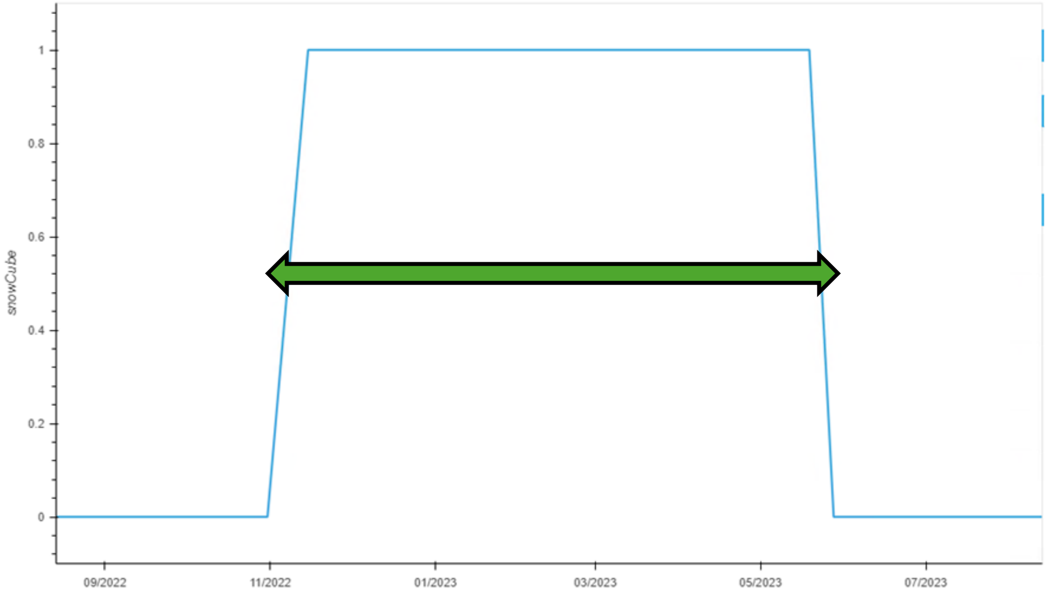
snow_end example for single pixel time-series from snow cube. snow_end will occur halfway along uncertainty period.



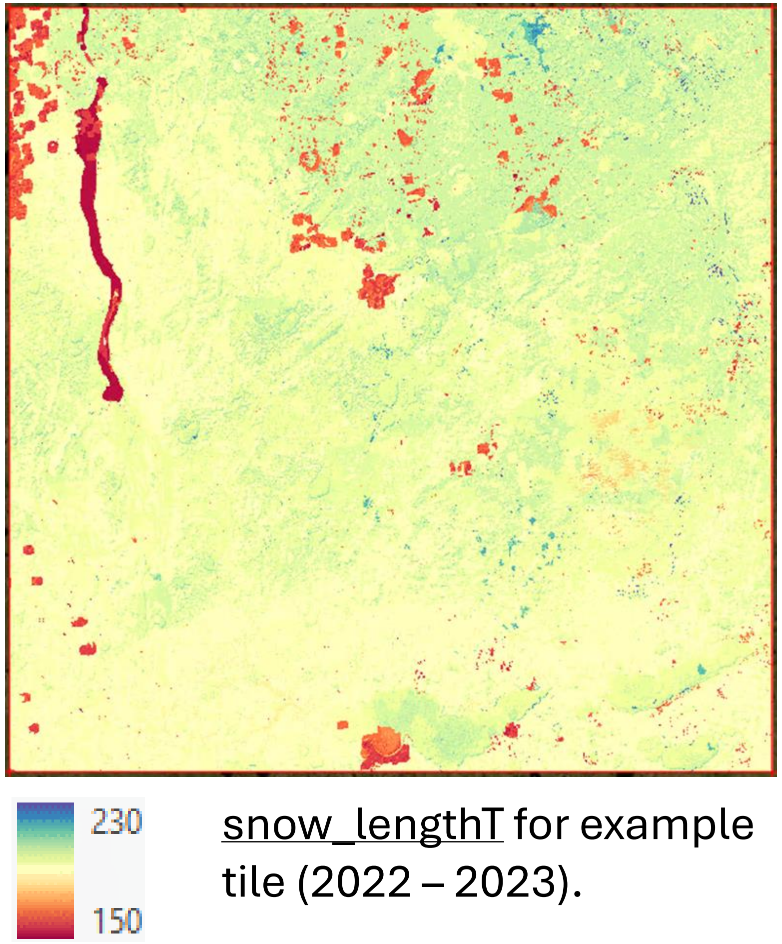
snow_endL for example tile (2022 – 2023). 120: May 10, 150: May 30.

6. Snow cube to snow dynamics (4)

- <u>snow_length</u>
  - Definition: Number of days with snow in total (T) or in the biggest (B) snow period in winter year
  - Pixel-level Calculation:
    - Number of days from snow start to end (inclusive) for one period, with T being the sum of all period lengths



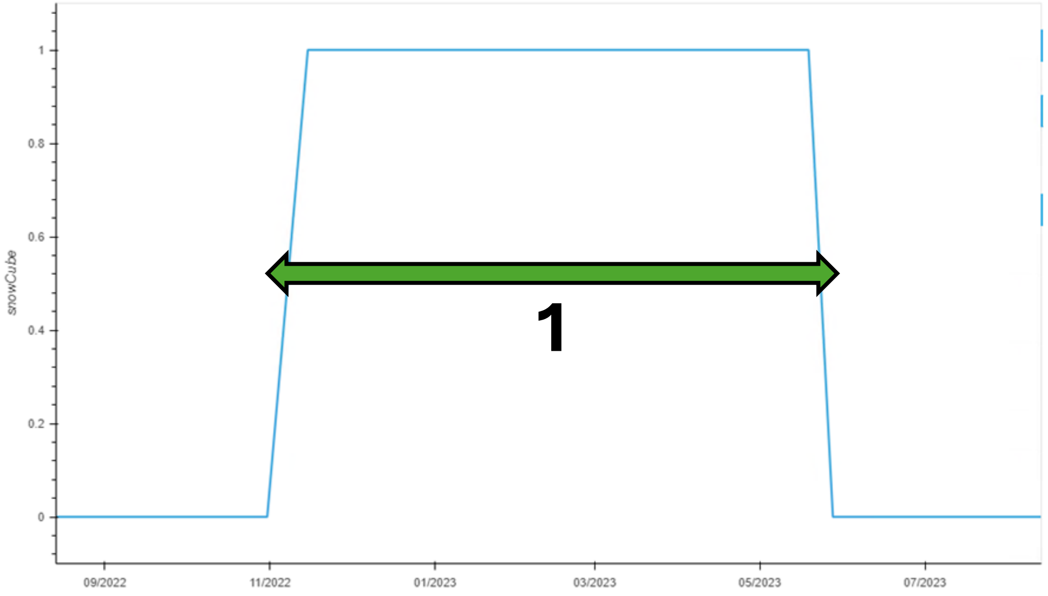
<u>snow_length</u> example for single pixel time-series from snow cube. <u>snow_length</u> is the sum of days from halfway points in edge uncertainty periods.



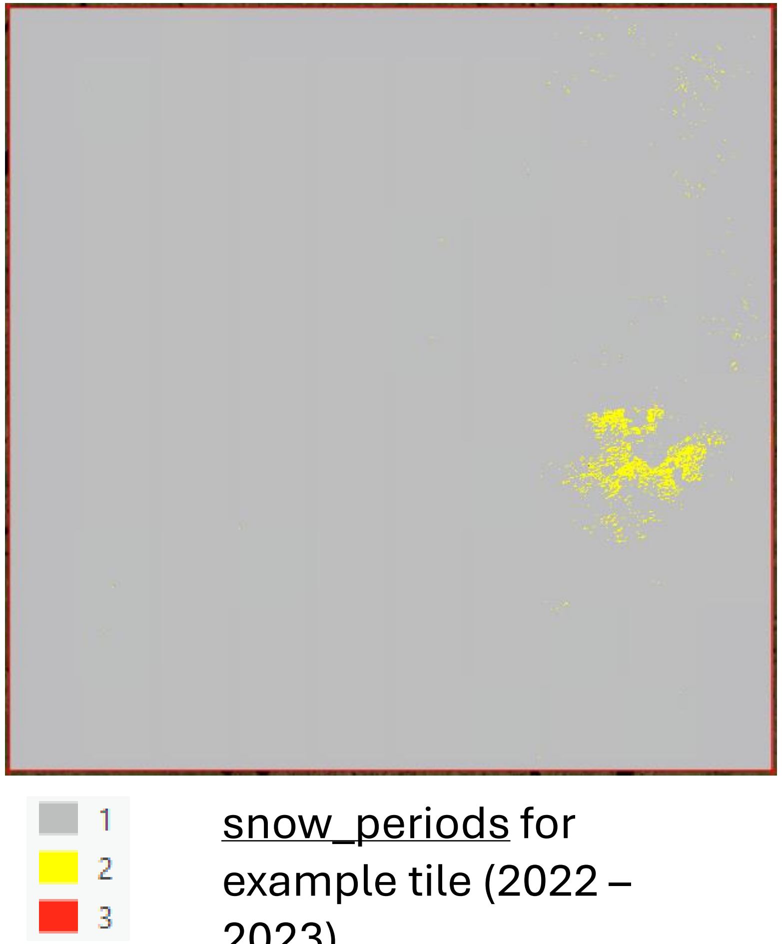
<u>snow_lengthT</u> for example tile (2022 – 2023).

6. Snow cube to snow dynamics (5)

- snow_periods
  - Definition: Number of separated snow periods during the winter year
  - Pixel-level Calculation:
    - Number of peaks identified, with edge cases accounted for



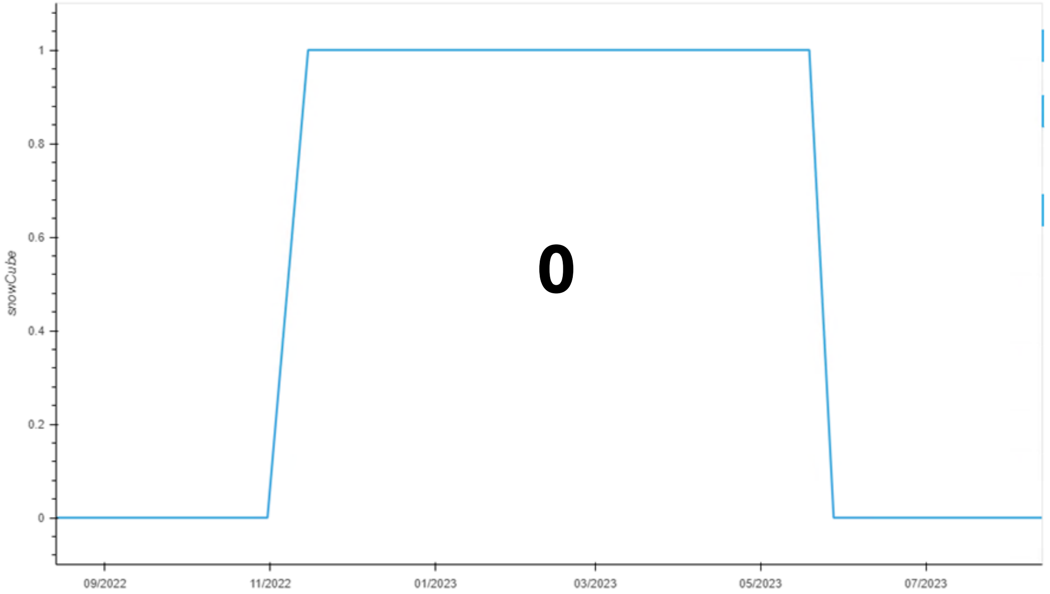
snow_periods example for single pixel time-series from snow cube.



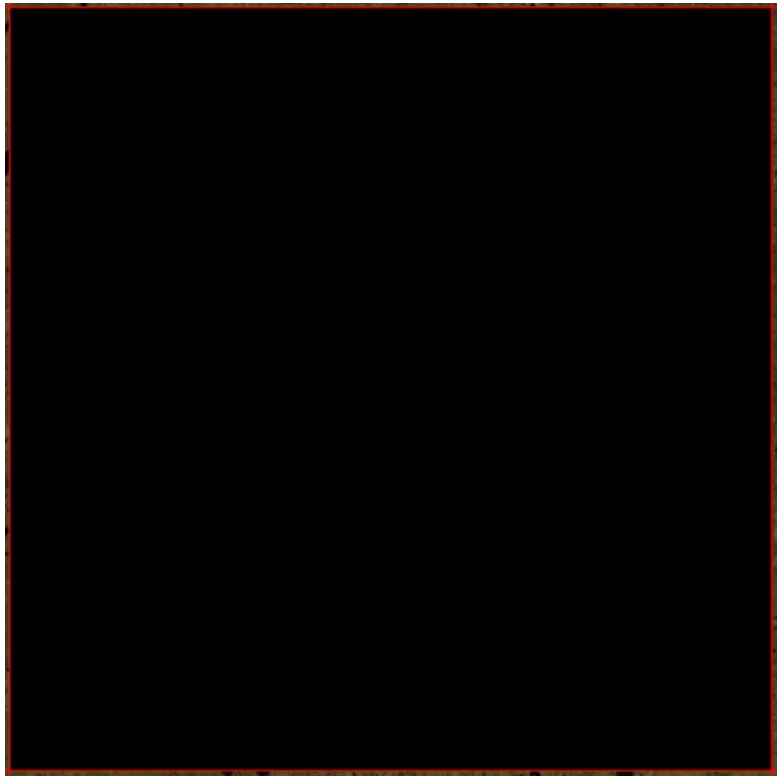
snow_periods for example tile (2022 – 2023).

6. Snow cube to snow dynamics (6)

- snow_status
  - Definition (pixel-level calculation):
    - 0: Seasonal snow (all other scenarios)
    - 1: Perennial snow (lengthT >= winter year length)
    - 2: Inconsistent perennial snow (snow observed within a week of winter year boundary)
    - 3: Snow free (lengthT = 0)
    - 4: Ephemeral snow (lengthB <= 7)



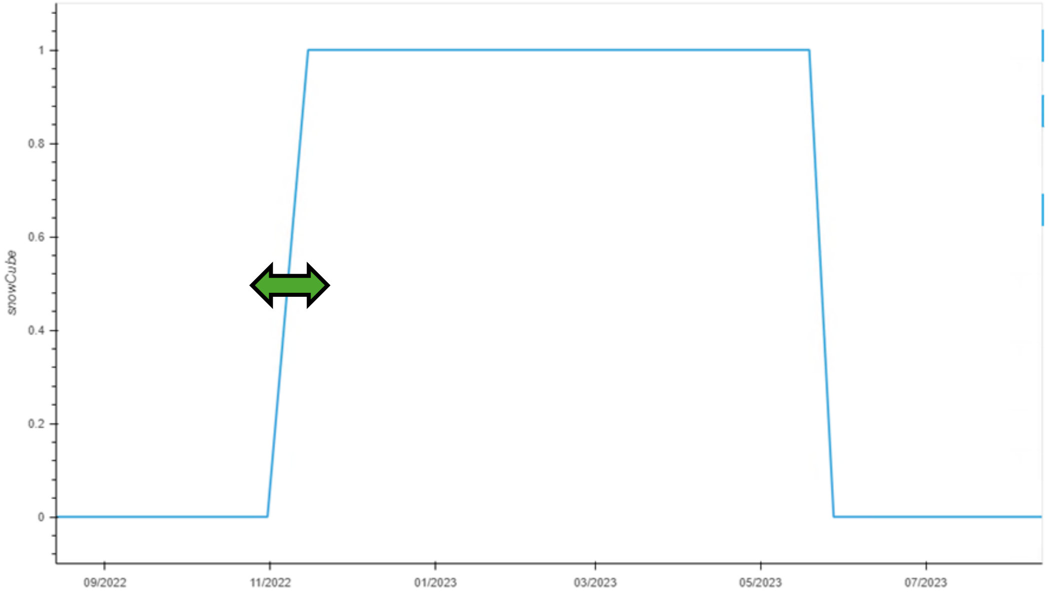
snow_status example for single pixel time-series from snow cube.



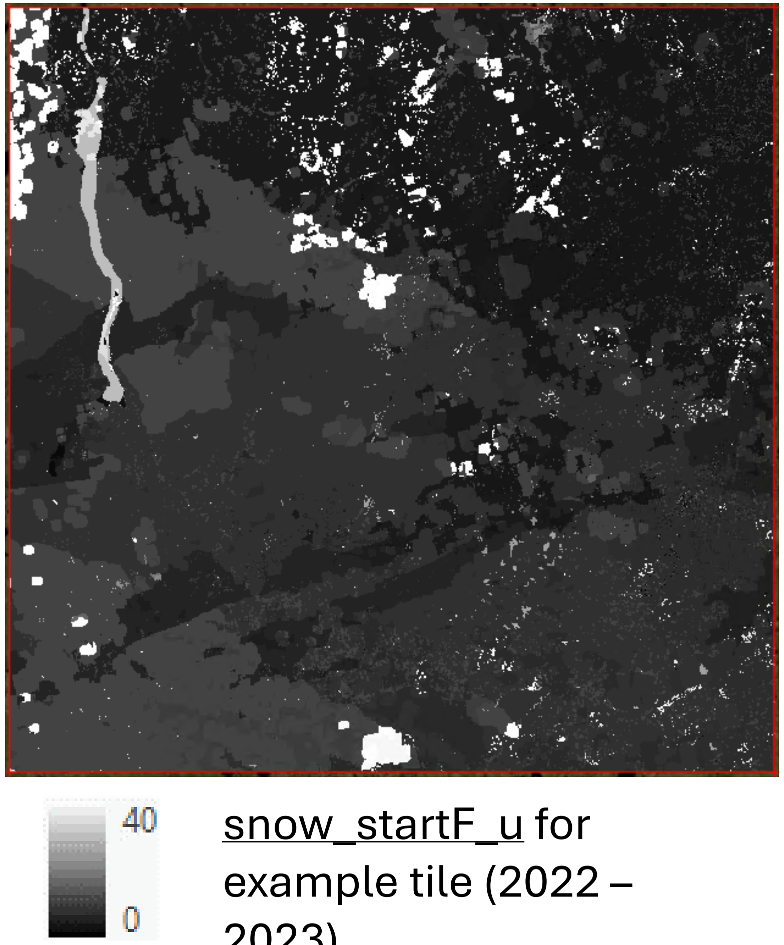
snow_status for example tile (2022 – 2023). All pixels = 0.

6. Snow cube to snow dynamics (7)

- snow_start_u
  - Definition: Uncertainty (± days) of snow start (F or B)
  - Pixel-level Calculation:
    - Days between left base of peak (previous non-snow) and next (first snow) observation, divided by 2



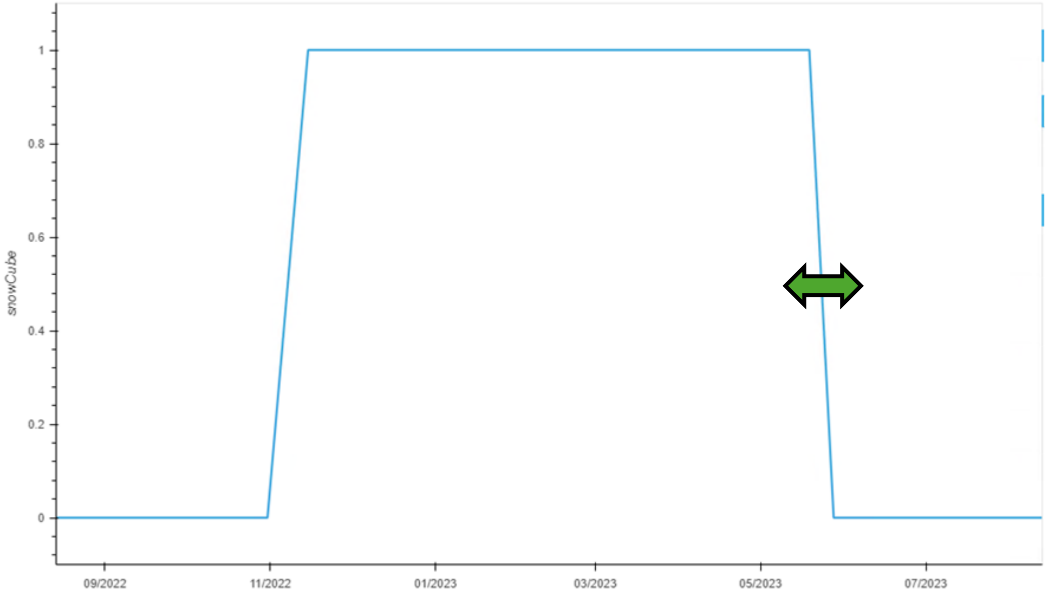
snow_start_u example for single pixel time-series from snow cube.



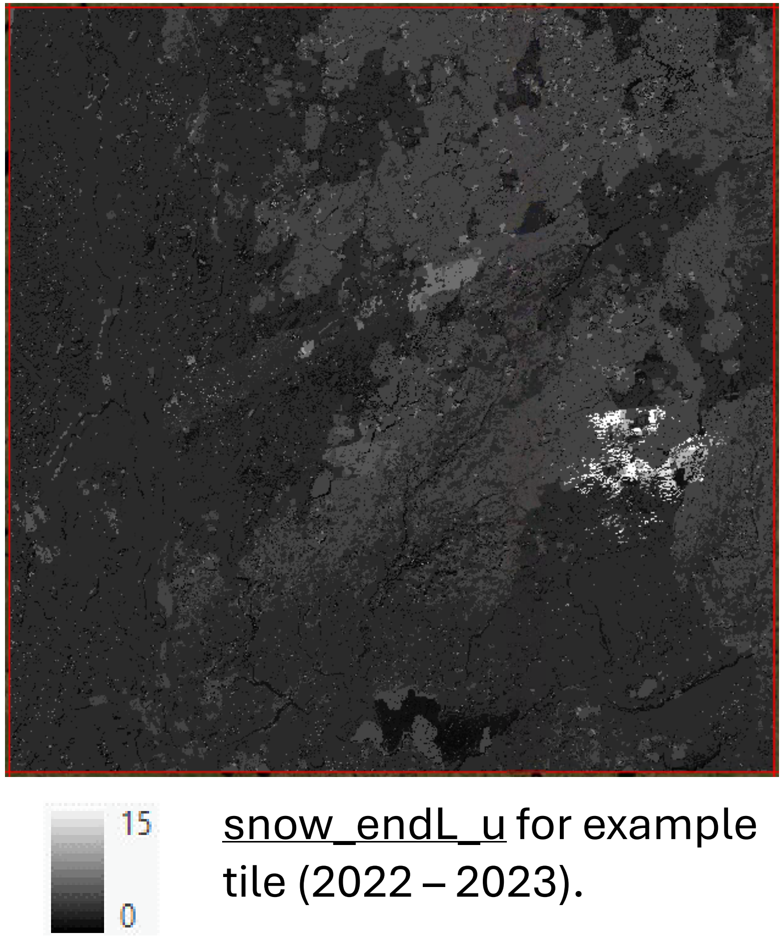
snow_startF_u for example tile (2022 – 2023).

6. Snow cube to snow dynamics (8)

- snow_end_u
  - Definition: Uncertainty (± days) of snow end (L or B)
  - Pixel-level Calculation:
    - Days between right base of peak (next non-snow) and previous (last snow) observation, divided by 2



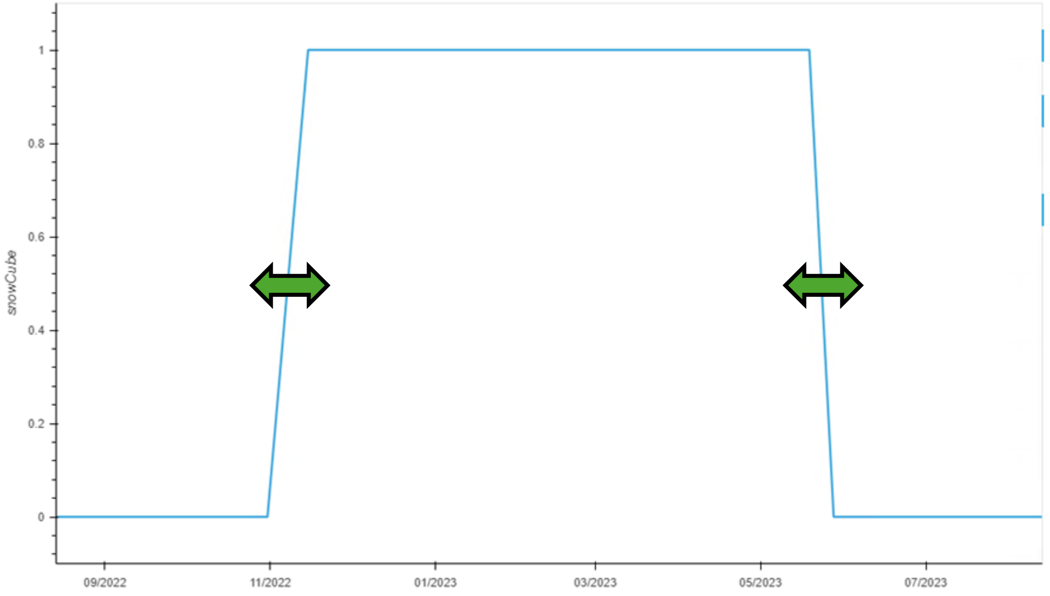
snow_end_u example for single pixel time-series from daily snow cube.



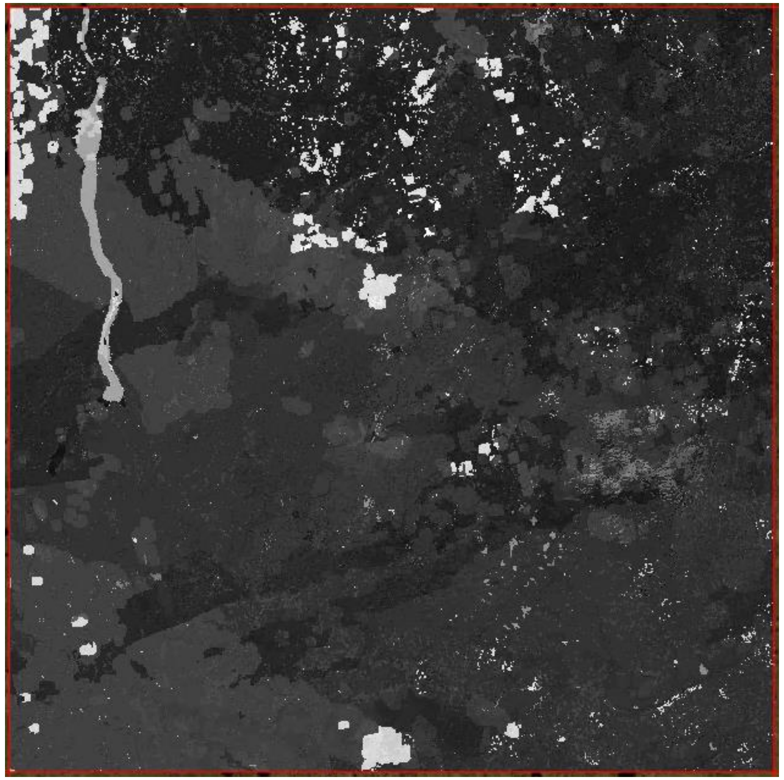
snow_endL_u for example tile (2022 – 2023).

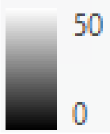
6. Snow cube to snow dynamics (9)

- snow_length_u
  - Definition: Uncertainty (± days) of snow length (T or B)
  - Pixel-level Calculation:
    - Sum of start_u and end_u for one period, with T being the sum of all period uncertainties



snow_length_u example for single pixel time-series from snow cube.





snow_lengthT_u for example tile (2022 – 2023).

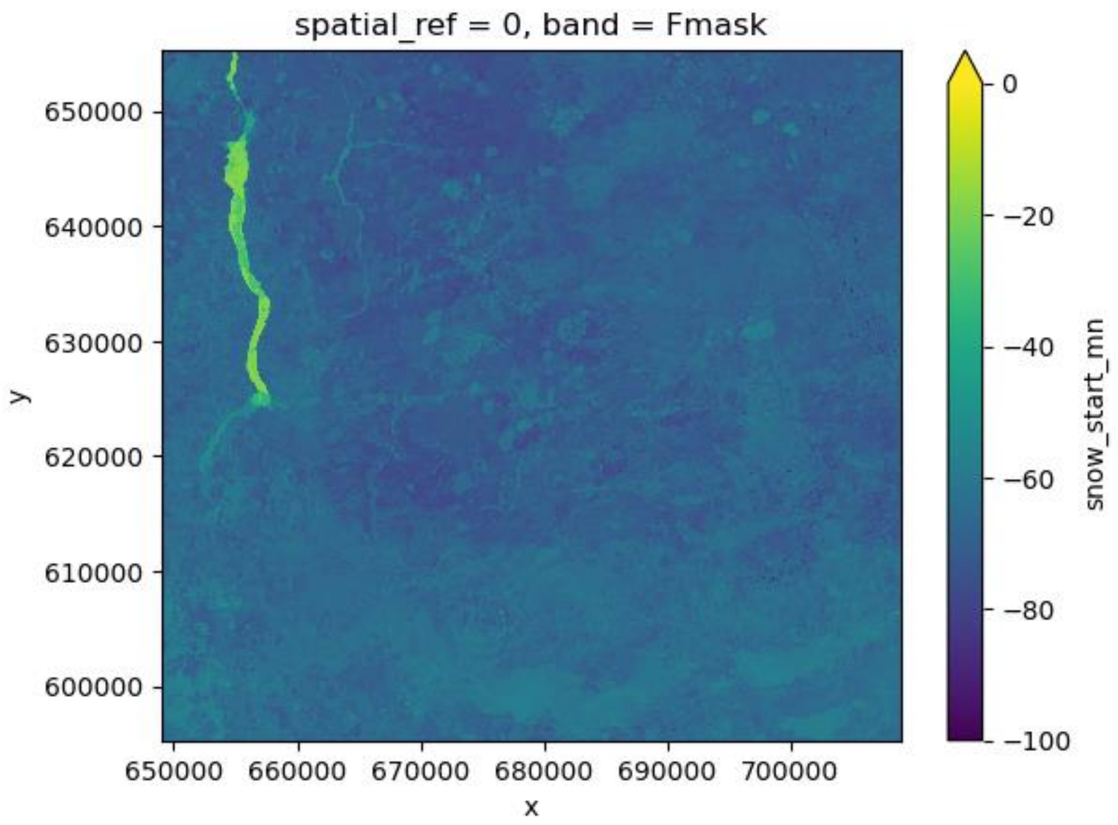
7. Snow dynamics merged to interannual (1)

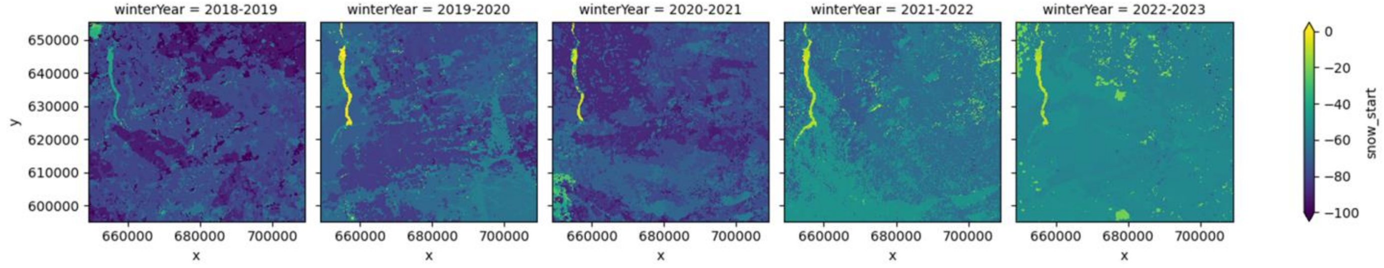
API functions:
- Snow_Utils.interannualSnowDynamics()

- Open winter year snow dynamics products as dataset
  - xarray's open_mfdataset works well for this
- Define key inputs for API function:
  - min_count: Minimum number of winter years with a value for each pixel to return a value for given product
    - Default: half rounded up (e.g., 3 for 5 winter years)
    - Matters mostly for snow_start and snow_end when pixels are classified as perennial or snow free
  - products: Snow dynamics products of interest (start, end, length, periods, status)
  - uncertainty: Snow dynamics uncertainty products of interest (start_u, end_u, length_u)
  - form: Form of interannual merging to apply
    - Default: weighted mean (by uncertainty and implausibility) for snow_start, snow_end, snow_length and their uncertainties; regular mean for snow_periods; % perennial and % snow free for snow_status
  - implausible_snow: Julian days between which it is implausible for there to be snow cover
    - Default: Applied from week after IMS's send_l to week before sstart_f for each tile
  - sd: Whether to create interannual standard deviation products for snow_start, snow_end, snow_length
    - Default: Don't create
  - quality: Whether to create interannual mean quality products for snow_start, snow_end, snow_length
    - Default: Create (requires weighted mean)
    - Calculated as weighted mean of weights (uncertainty and implausibility)
  - best_value: Whether or not to create interannual best value products for snow_start, snow_end, snow_length and their uncertainties
    - Default: Don't create (requires weighted mean)
    - Outputs best value (bv), year of best value (bvy), and quality of best value (bvq, using weighted mean quality weights)

7. Snow dynamics merged to interannual (2)

- snow_start_mn (weighted mean, F and B)
  - Remove values where number of valid (non-NaN) values across winter years < min_count
    - i.e., If 3/5 winter years are NaN, set remaining 2 to NaN
  - Generate weights for each winter year
    - Uncertainty (u): 2.718 ** (-0.046 * x)
      - More details in (5)
    - implausibility (i): 2.718 ** (-0.046 * x)
      - Implausible snow dates (i1)
      - Implausible snow dynamics variation (i2)
      - More details in (5)
    - Combine (50% u, 50% i)
      - (u * 0.5) + (i1 * 0.25) + (i2 * 0.25)
  - For each pixel, calculate weighted mean using combined weights

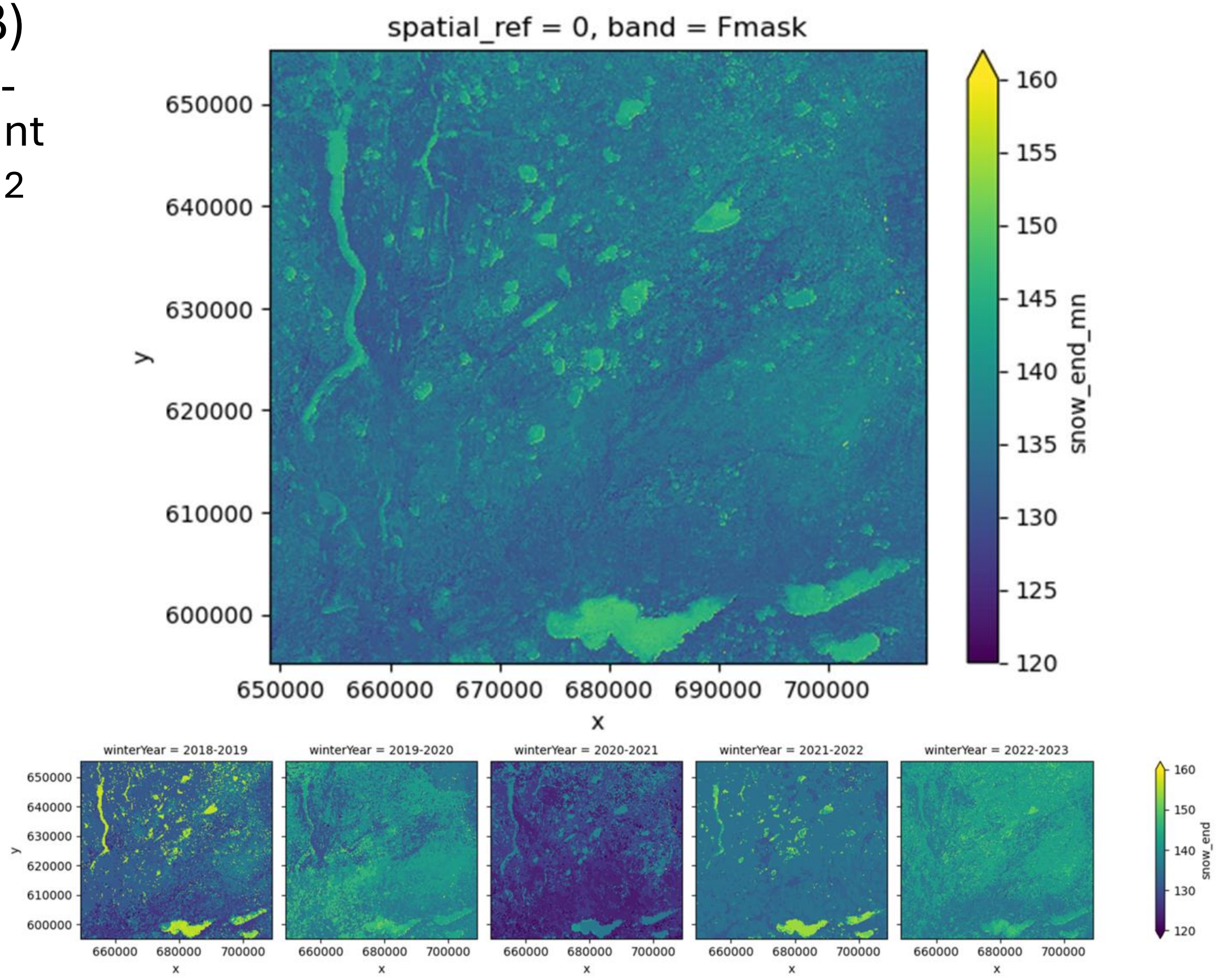




API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (3)

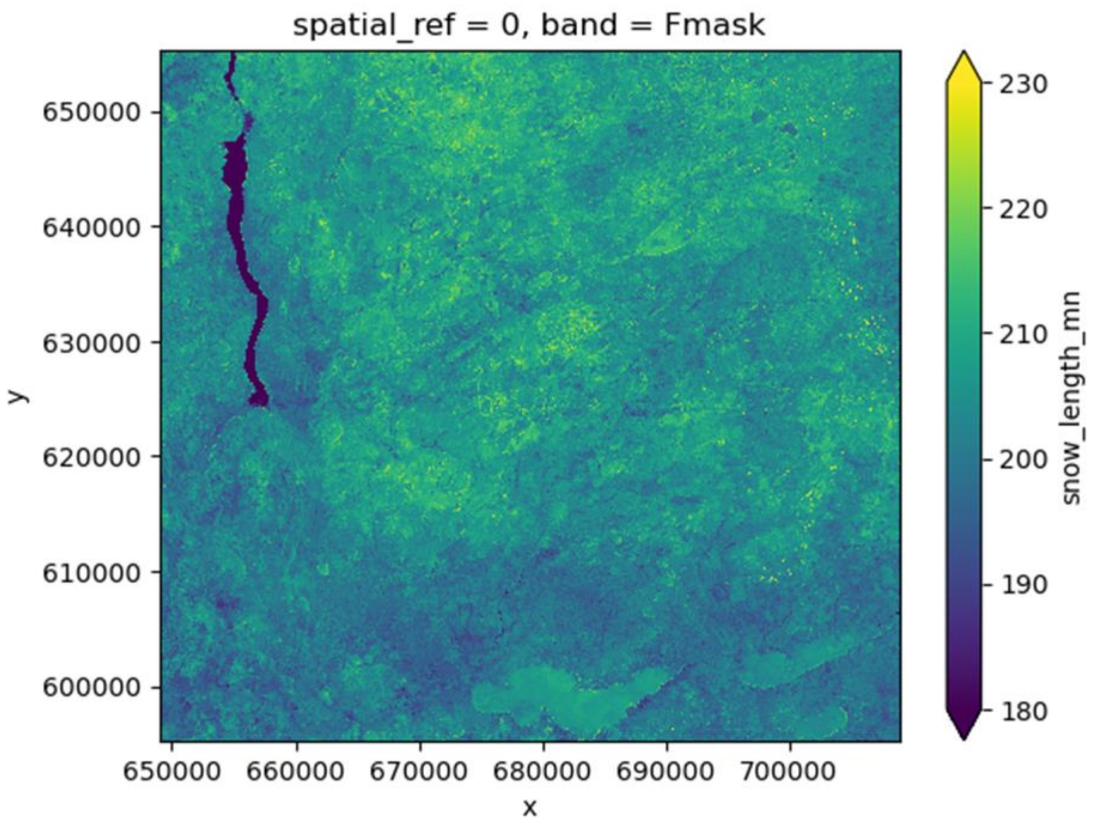
- <u>snow_end_mn</u> (weighted mean, L and B)
  - Remove values where number of valid (non-NaN) values across winter years < min_count
    - i.e., If 3/5 winter years are NaN, set remaining 2 to NaN
  - Generate weights for each winter year
    - Uncertainty (u): 2.718 ** (-0.046 * x)
      - More details in (5)
    - implausibility (i): 2.718 ** (-0.046 * x)
      - Implausible snow dates (i1)
      - Implausible snow dynamics variation (i2)
      - More details in (5)
    - Combine (50% u, 50% i)
      - (u * 0.5) + (i1 * 0.25) + (i2 * 0.25)
  - For each pixel, calculate weighted mean using combined weights

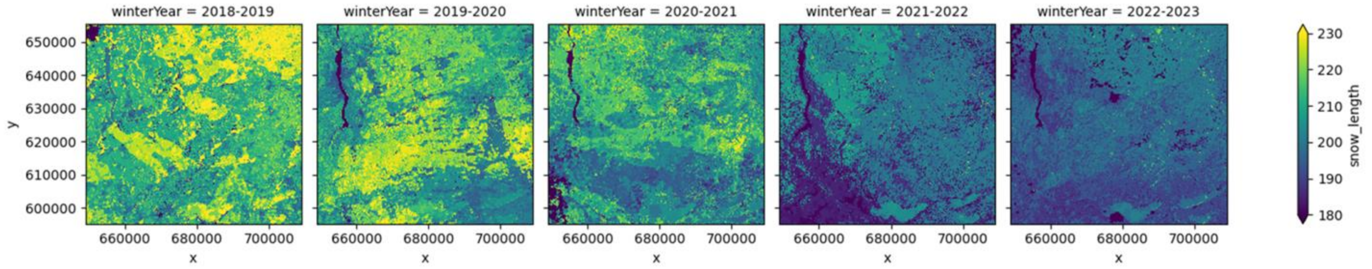


API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (4)

- <u>snow_length_mn</u> (weighted mean, T and B)
  - Remove values where number of valid (non-NaN) values across winter years < min_count
    - i.e., If 3/5 winter years are NaN, set remaining 2 to NaN
  - Generate weights for each winter year
    - Uncertainty (u): 2.718 ** (-0.023 * x)
      - More details in (5)
    - implausibility (i): 2.718 ** (-0.046 * x)
      - Implausible snow dates (i1)
      - Implausible snow dynamics variation (i2)
      - More details in (5)
    - Combine (50% u, 50% i)
      - (u * 0.5) + (i1 * 0.25) + (i2 * 0.25)
  - For each pixel, calculate weighted mean using combined weights

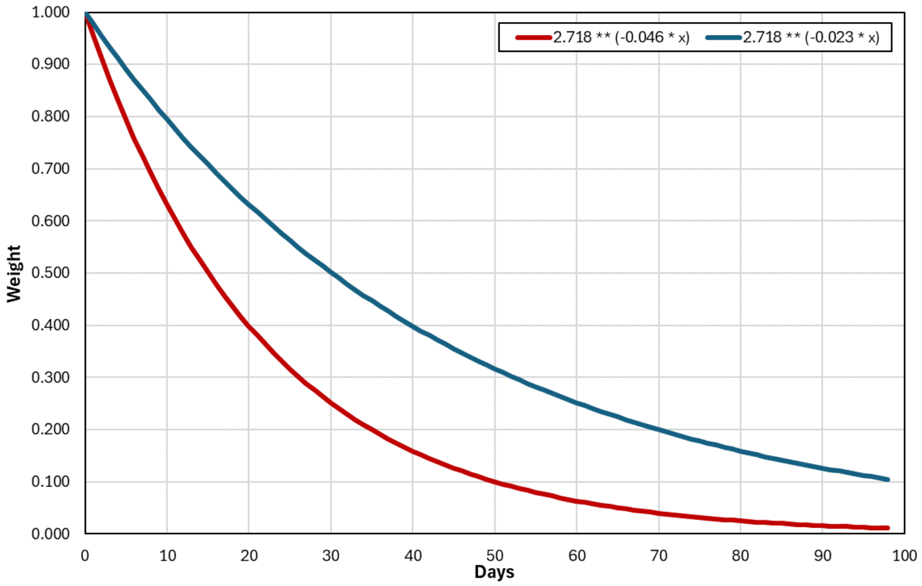




API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (5)

- Weighted mean calculation
  - For each pixel and winter year:
    - Uncertainty: 50% weight (u)
      - snow_start and snow_end: 2.718 ** (-0.046 * snow_start/end_u)
      - snow_length: 2.718 ** (-0.023 * snow_length_u)
    - Implausibility: 50% weight
      - Implausible snow dates: 25% (i1)
        - Identify likely implausible values (absolute value outside normal range for snow dynamic statistic)
          - snow_start: Before early start date (before end of implausible_snow) and after late start date (at least 95th percentile of snow_start)
          - snow_end: After late end date (after start of implausible_snow) and before early end date (at and below 5th percentile of snow_end)
          - snow_length: Too long (longer than 365 – implausible_snow period length) and too short (at and below 5th percentile of snow_length)
        - Calculate weight: 2.718 ** (-0.046 * days outside)
      - Implausible snow dynamics variation: 25% (i2)
        - Calculate absolute difference between value and median of all values across winter years
        - Calculate weight: 2.718 ** (-0.046 * days from median)
    - Combine weights with weighted sum: (u * 0.5) + (i1 * 0.25) + (i2 * 0.25)
      - Note: All weights are between 0 and 1 after applying exponentials
  - Calculate weighted mean using combined weights across winter years

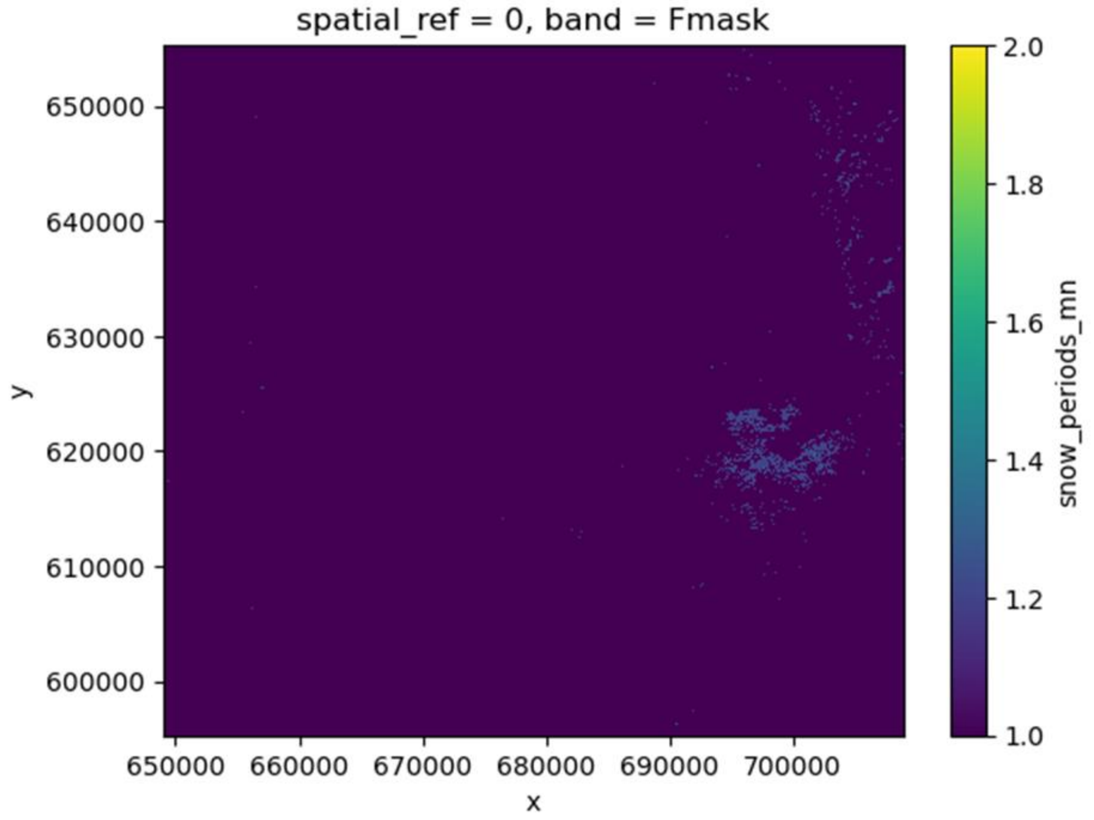


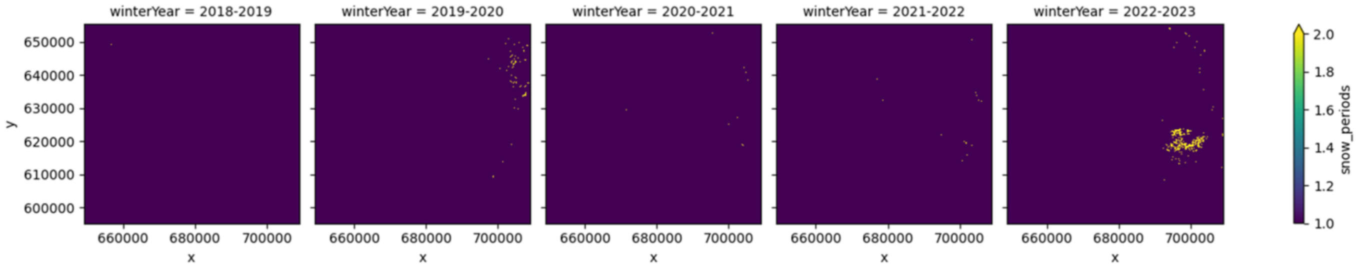
Visualization of equations for weight subsets (u, i1, i2). Red line is most common equation (50% weight at 15 days). Blue line used for snow_length uncertainty (50% weight at 30 days). Days: uncertainty days (u), days outside normal range (i1), days from median (i2).

API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (6)

- snow_periods_mn (mean)
  - Remove values where number of valid (non-NaN) values across winter years < min_count
    - i.e., If 3/5 winter years are NaN, set remaining 2 to NaN
  - Calculate mean

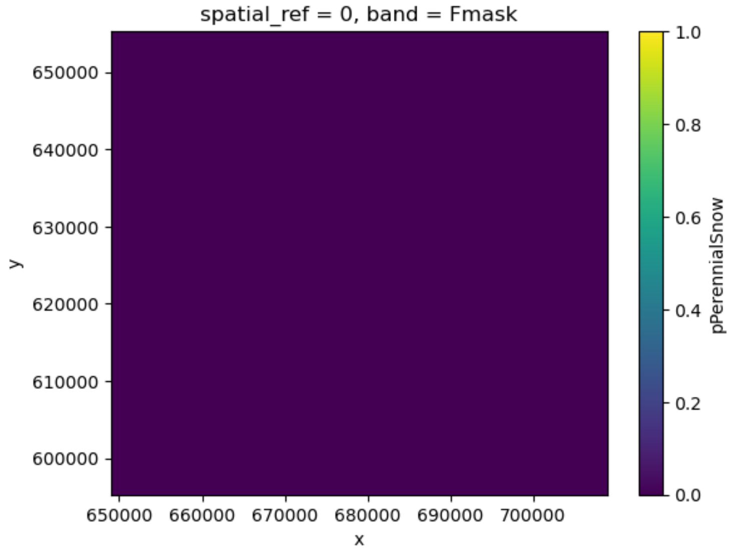


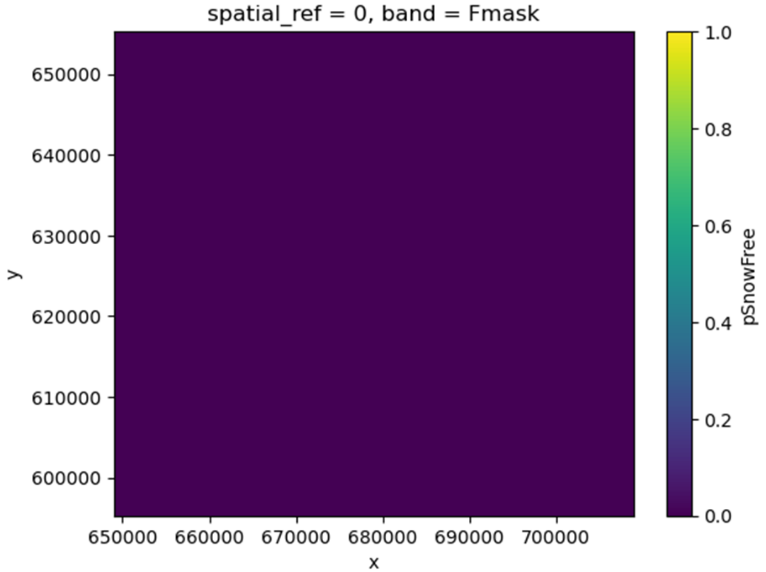


API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (7)

- <u>snow_status</u>
  - pPerennialSnow:
    - Reclassify so that perennial and inconsistent perennial = 1, and snow free = 0
    - Calculate the sum of all pixels reaching min_count, divided by number of valid (non-NaN) values across winter years * 100
      - i.e., sum = 3 (3 perennial winter years) / 5 valid values * 100 = 60%
  - pSnowFree:
    - Reclassify so that snow free = 1 and perennial and inconsistent perennial = 0
    - Calculate the sum of all pixels reaching min_count, divided by number of valid (non-NaN) values across winter years * 100
      - i.e., sum = 3 (3 snow free winter years) / 5 valid values * 100 = 60%

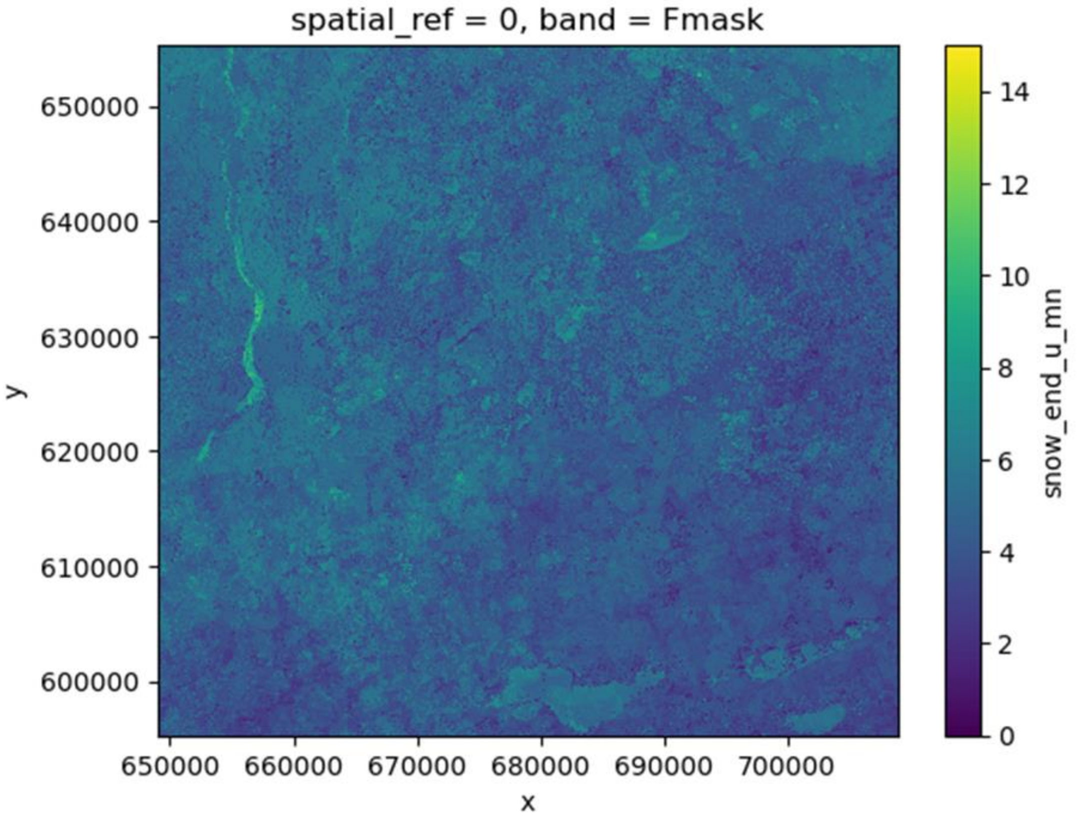


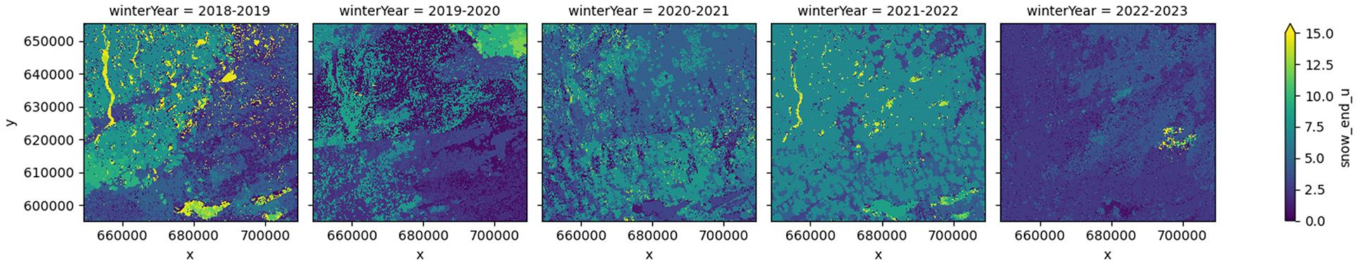


API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (8)

- Uncertainties: snow_start_u_mn, snow_end_u_mn, snow_length_u_mn (weighted mean, all versions)
  - Match NaNs with corresponding snow dynamic statistic
    - i.e., snow_start_u should have missing data for the same pixels as snow_start
  - Calculate weighted mean using corresponding combined weights
    - i.e., snow_start_u uses combined weights from snow_start

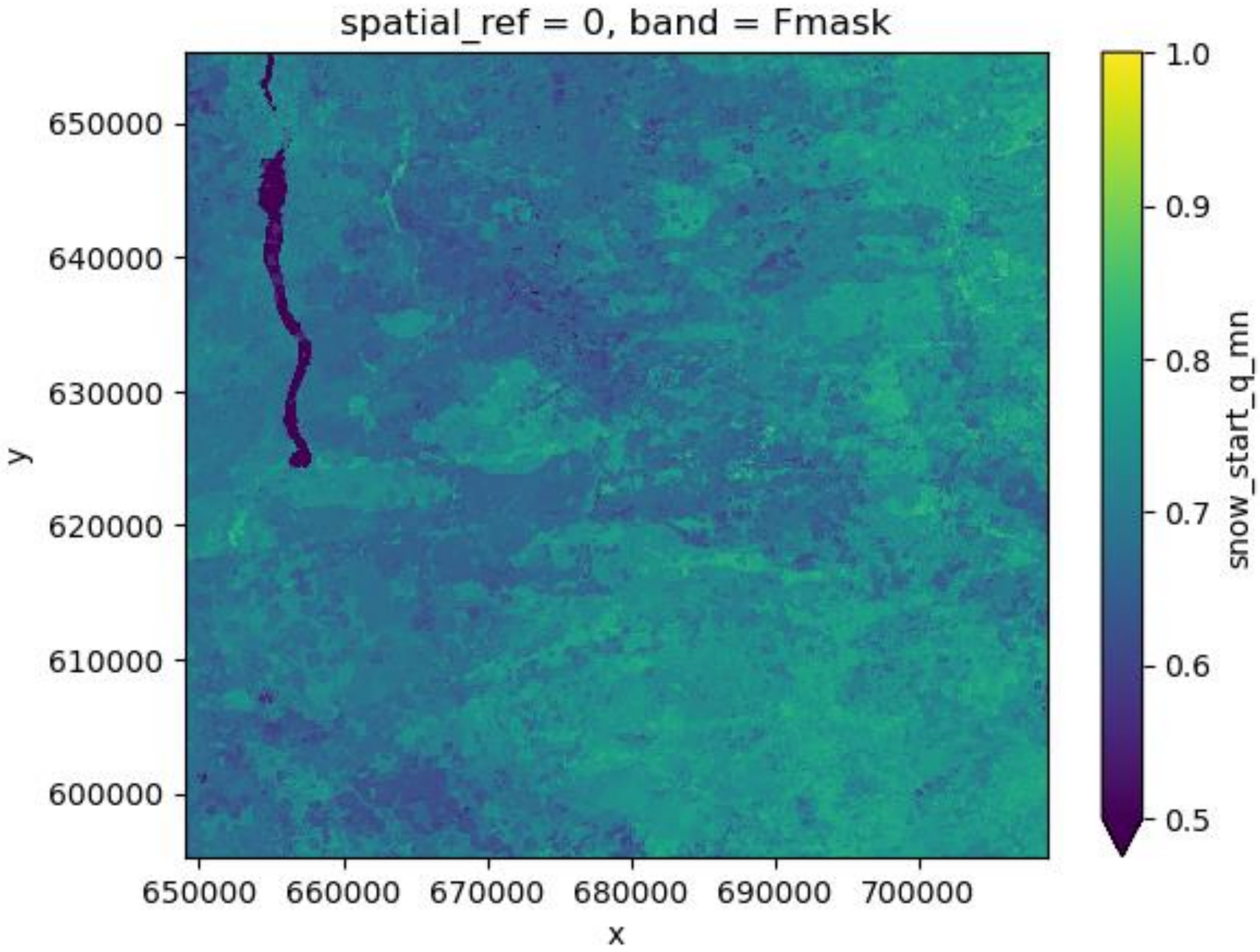




API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (9)

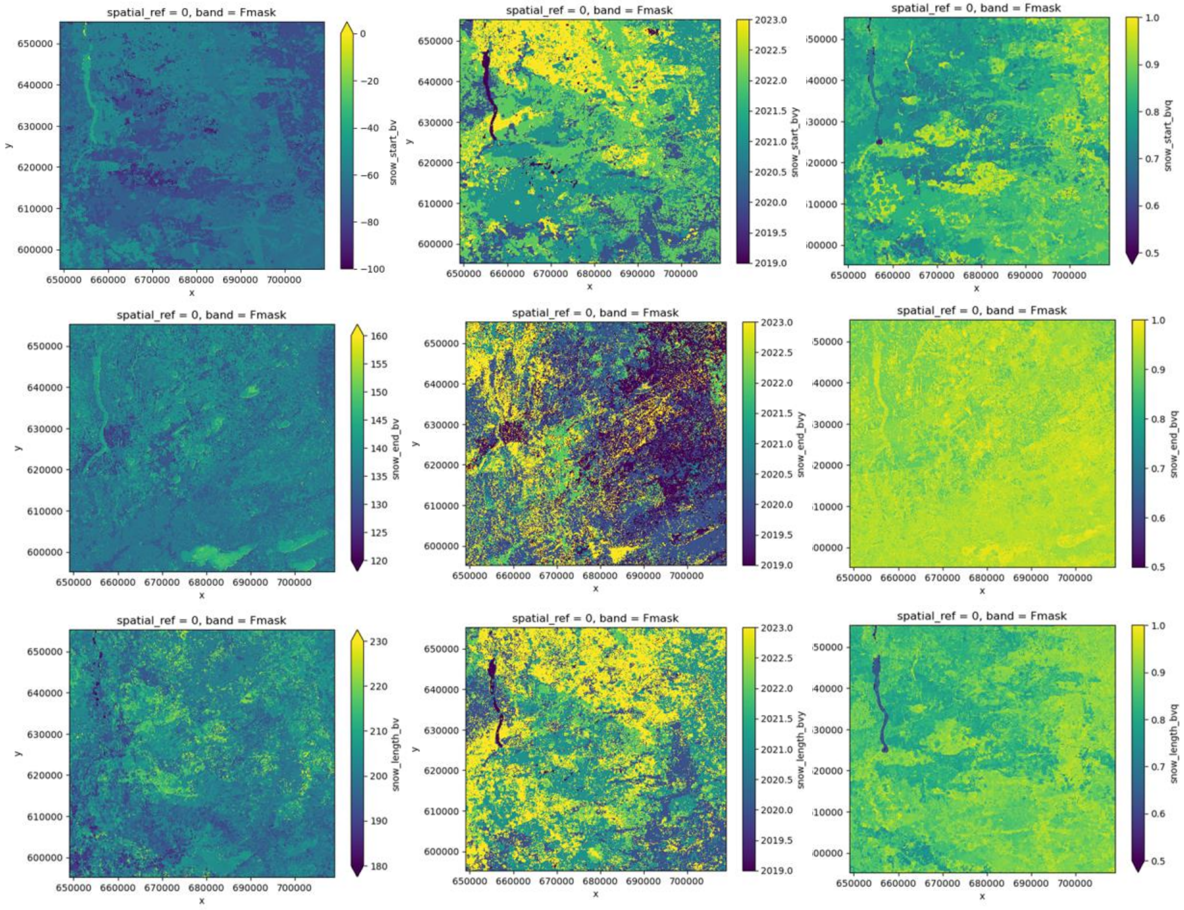
- Quality: snow_start_q_mn, snow_end_q_mn, snow_length_q_mn (weighted mean, all versions)
  - Calculate weighted mean of combined weights
    - i.e., snow_start_q_mn uses combined weights from snow_start



API functions:
- Snow_Utils.interannualSnowDynamics()

7. Snow dynamics merged to interannual (10)

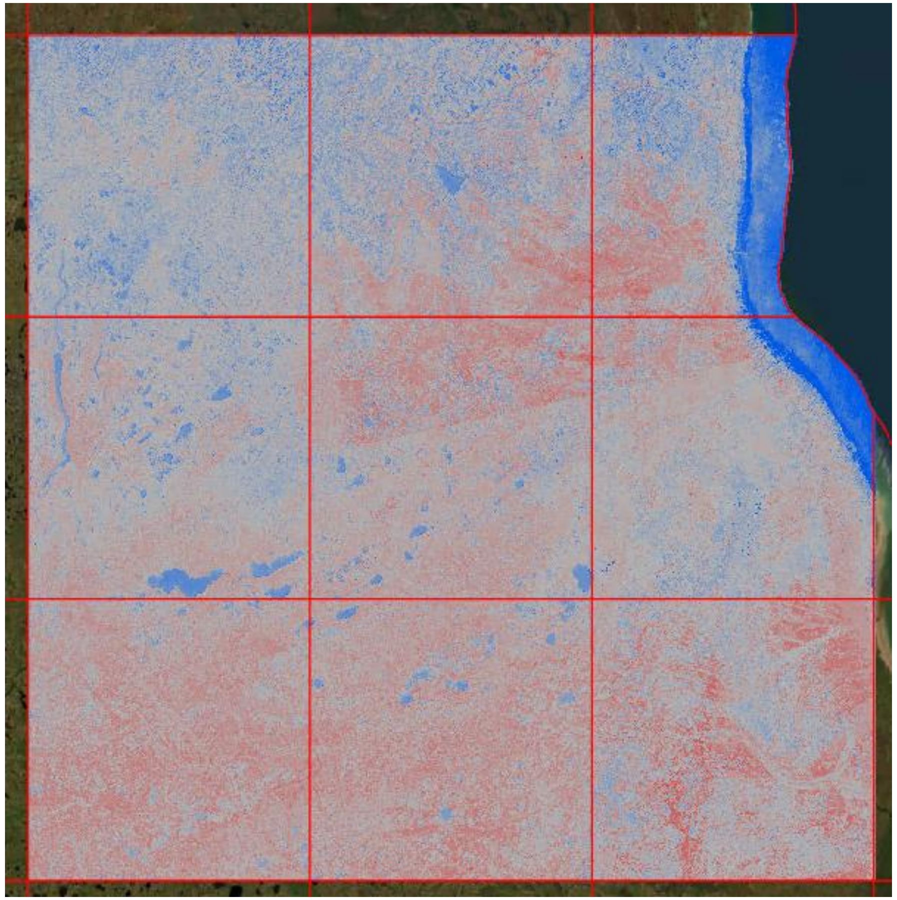
- Best value
  - Products: All versions
    - snow_start: snow_start_bv, snow_start_bvy, snow_start_bvq
    - snow_end: snow_end_bv, snow_end_bvy, snow_end_bvq
    - snow_length: snow_length_bv, snow_length_bvy, snow_length_bvq
  - For all pixels and winter years:
    - Find the index (0 start) with the highest weight
      - i.e., Return 3 if winter year 2021-2022 had the maximum combined weight
    - Select the product value corresponding to highest weight index (by)
      - i.e., Select snow_end value from 2021-2022 for highest weight = 3
    - Return the best value year by adding 2019 to 0 start index of highest weight (bvy)
      - i.e., 3 + 2019 = 2022 (second year of winter year)
    - Return best value quality in same manner as by (bvq)
      - i.e., Select snow_end combined weight value from 2021-2022 for highest weight = 3



API functions:
- Snow_Utils.interannualSnowDynamics()

8. Tile-based products merged

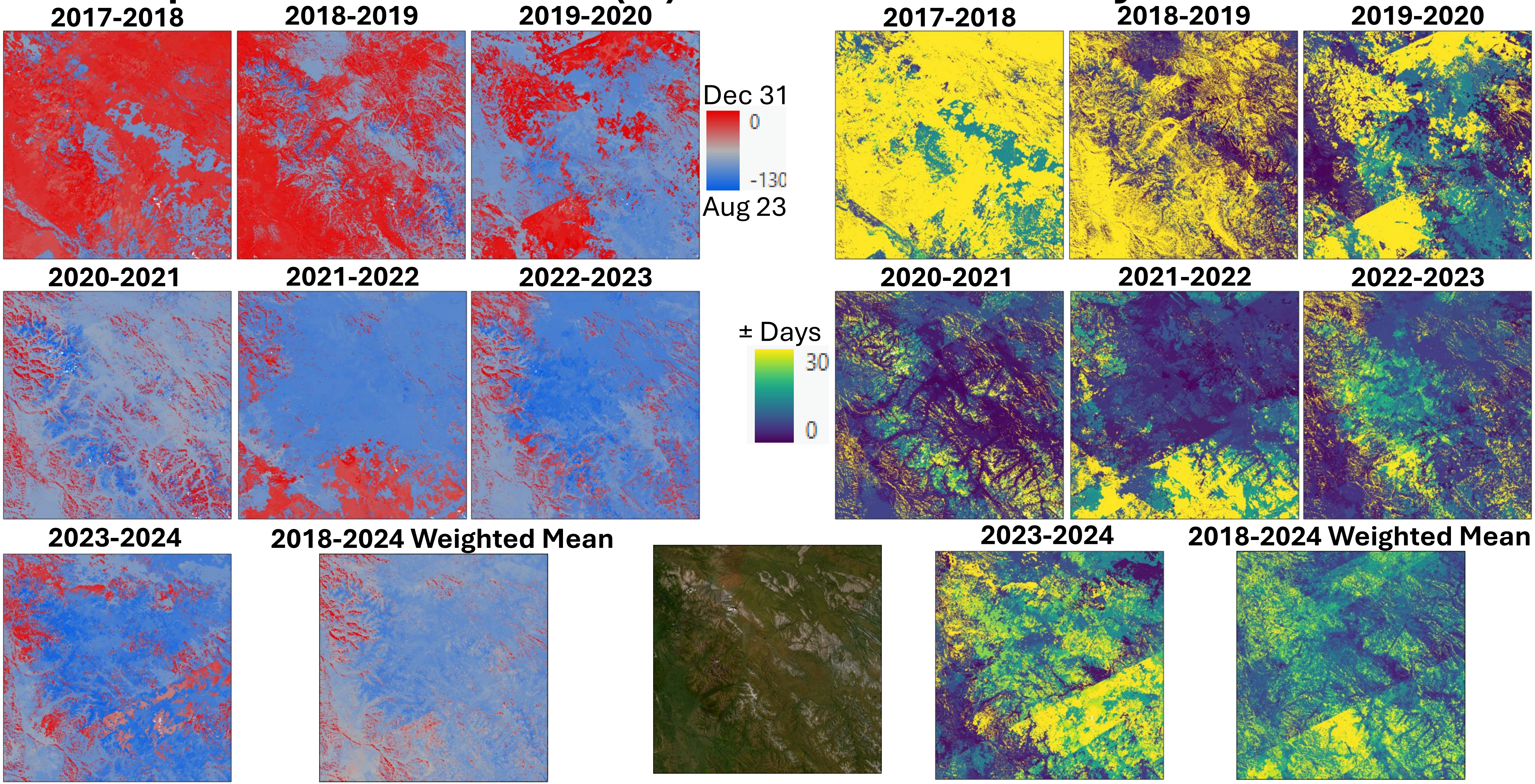
- Find all NetCDF files saved from previous steps based on input_ID
- Load together as single spatial dataset
  - As part of load function, remove edge along 2 overlapping sides (tiles have 1-pixel overlap)
- For each product (snow dynamics variable):
  - Clip to region of interest (e.g., Canada) if needed
  - Save to raster (TIF) with specified output_ID



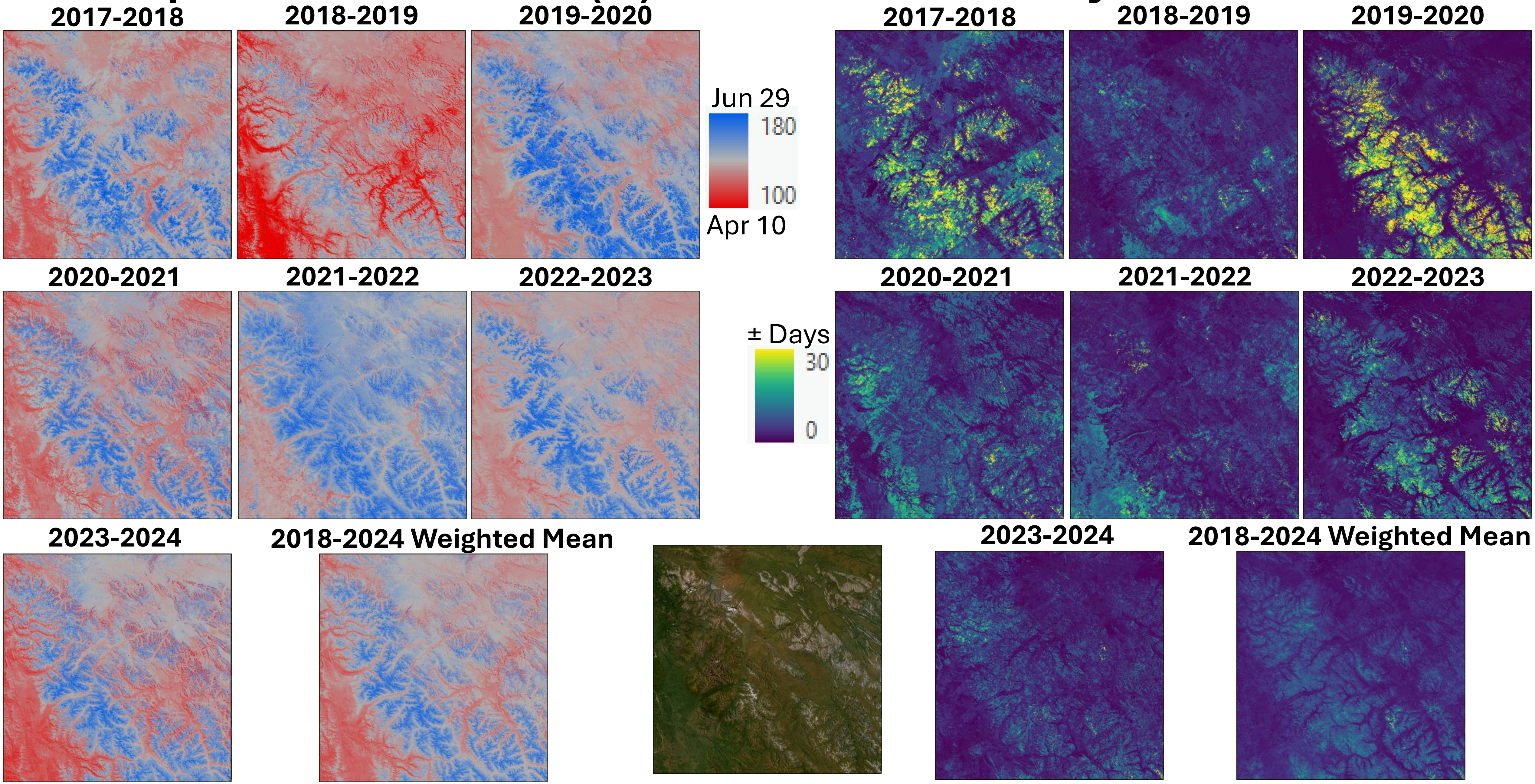
Example 3x3 tile merged snow_end_mn.

API functions:
- Snow_Utils. mergeTiledSnowDynamics()

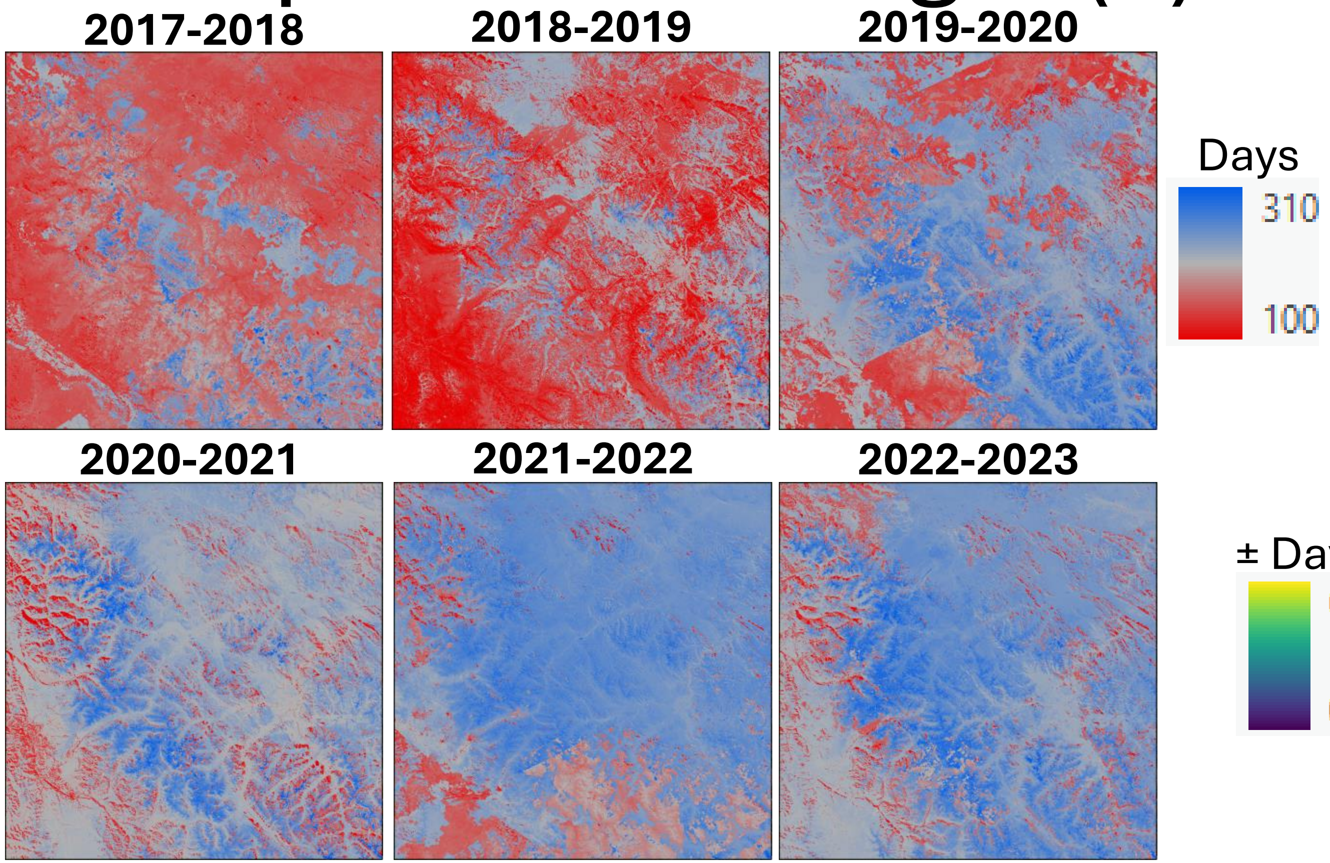
Example: Snow start (F) and uncertainty

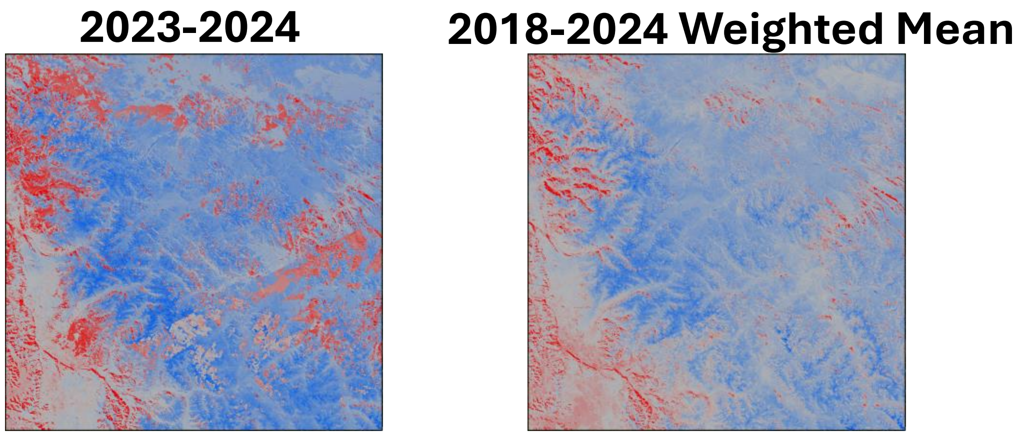


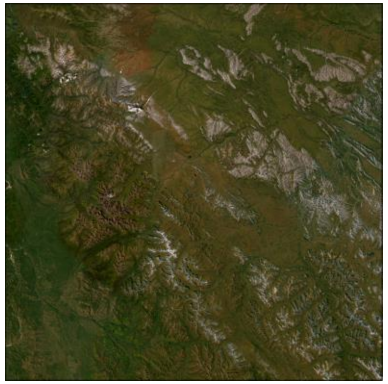
Example: Snow end (L) and uncertainty

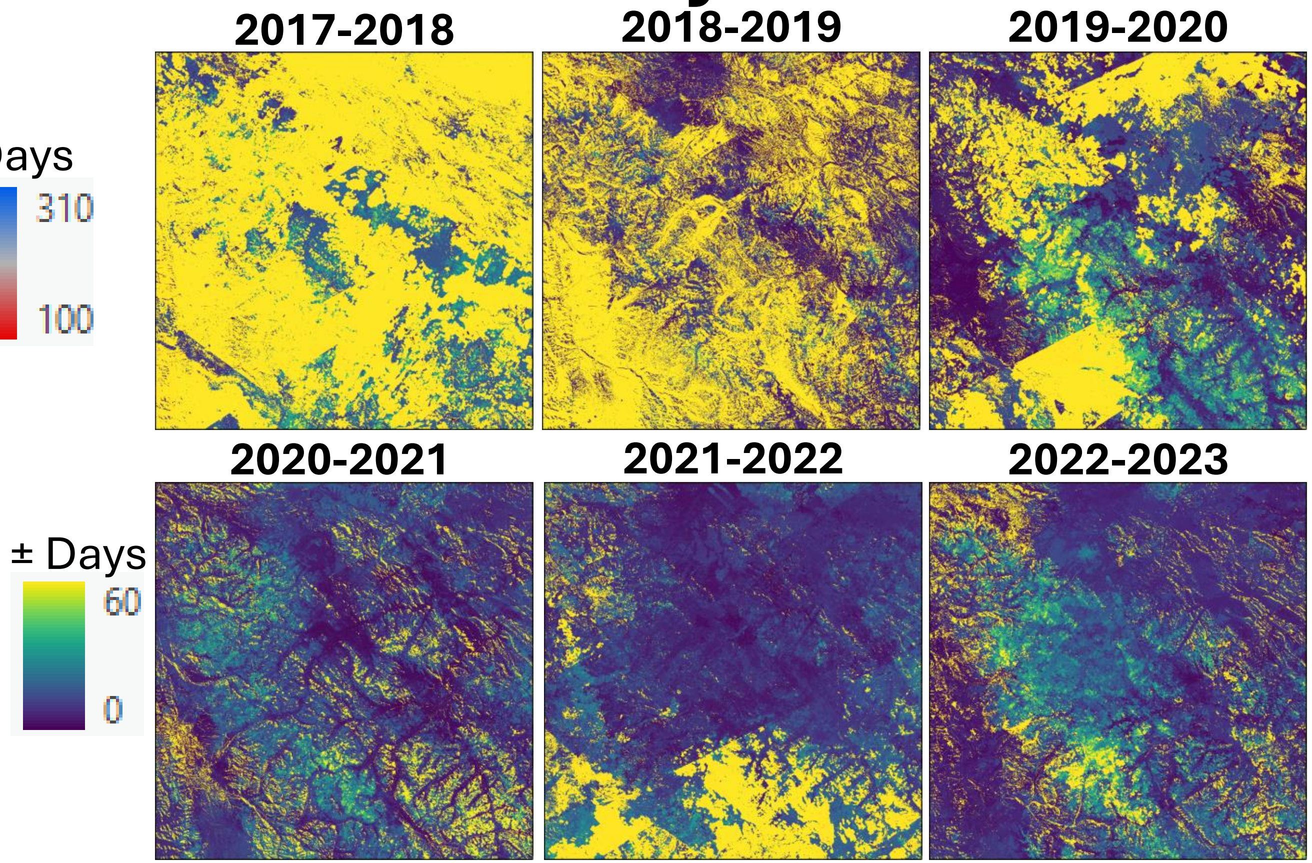


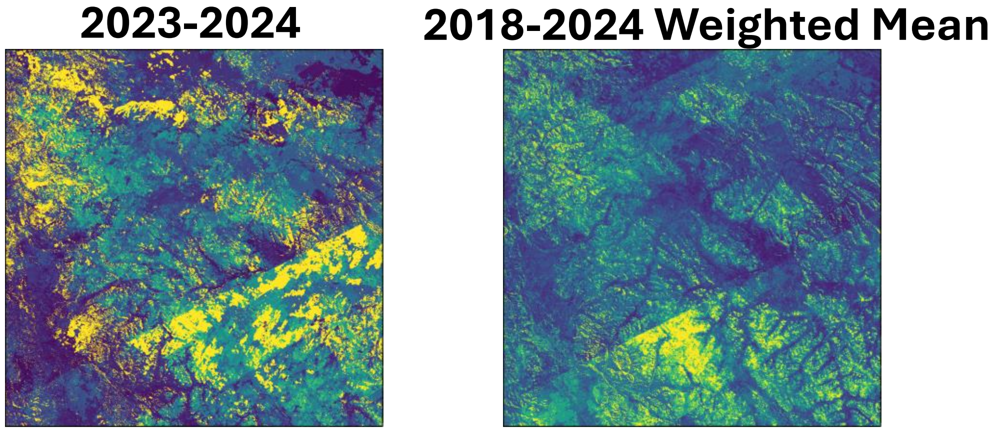
Example: Snow length (T) and uncertainty



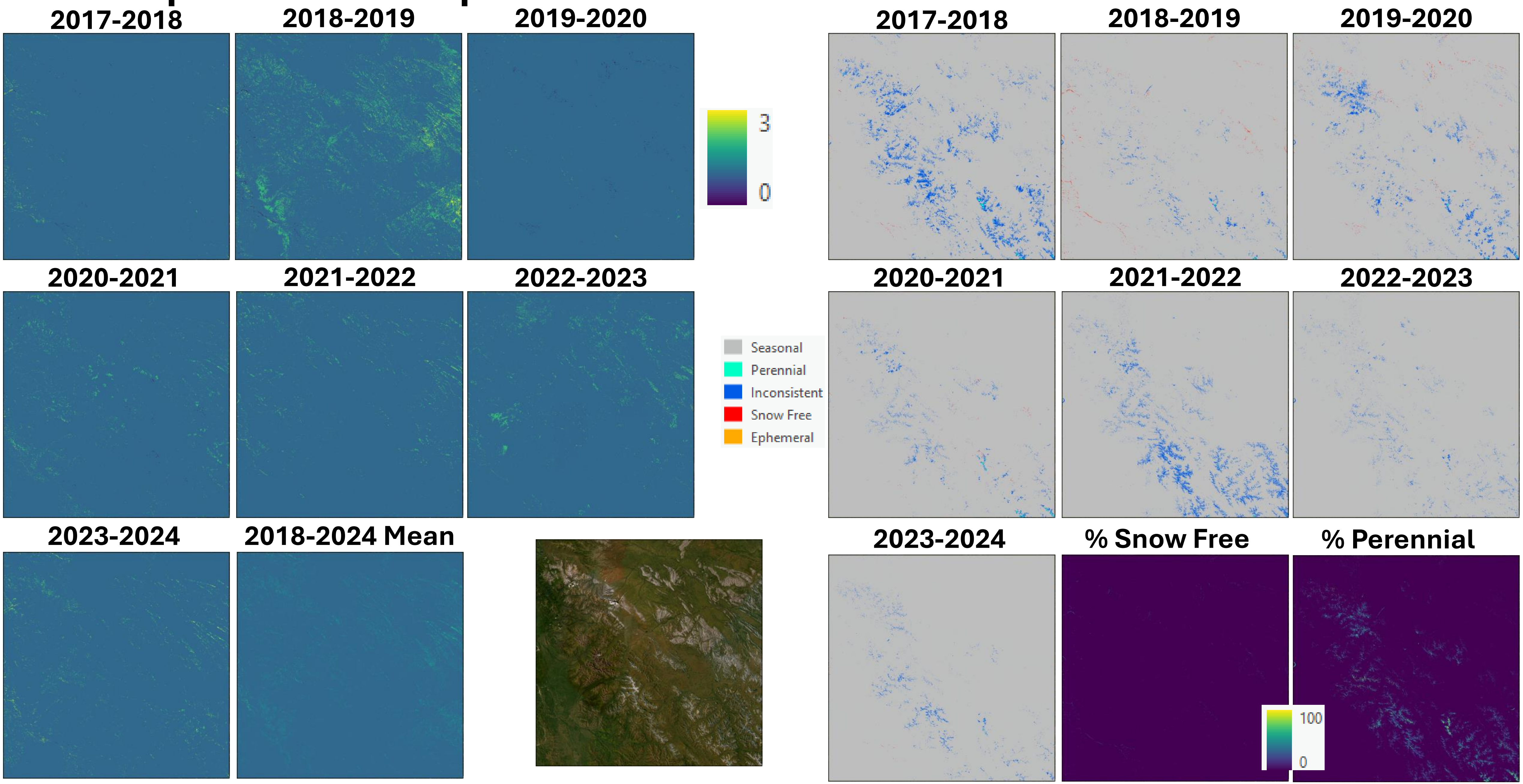




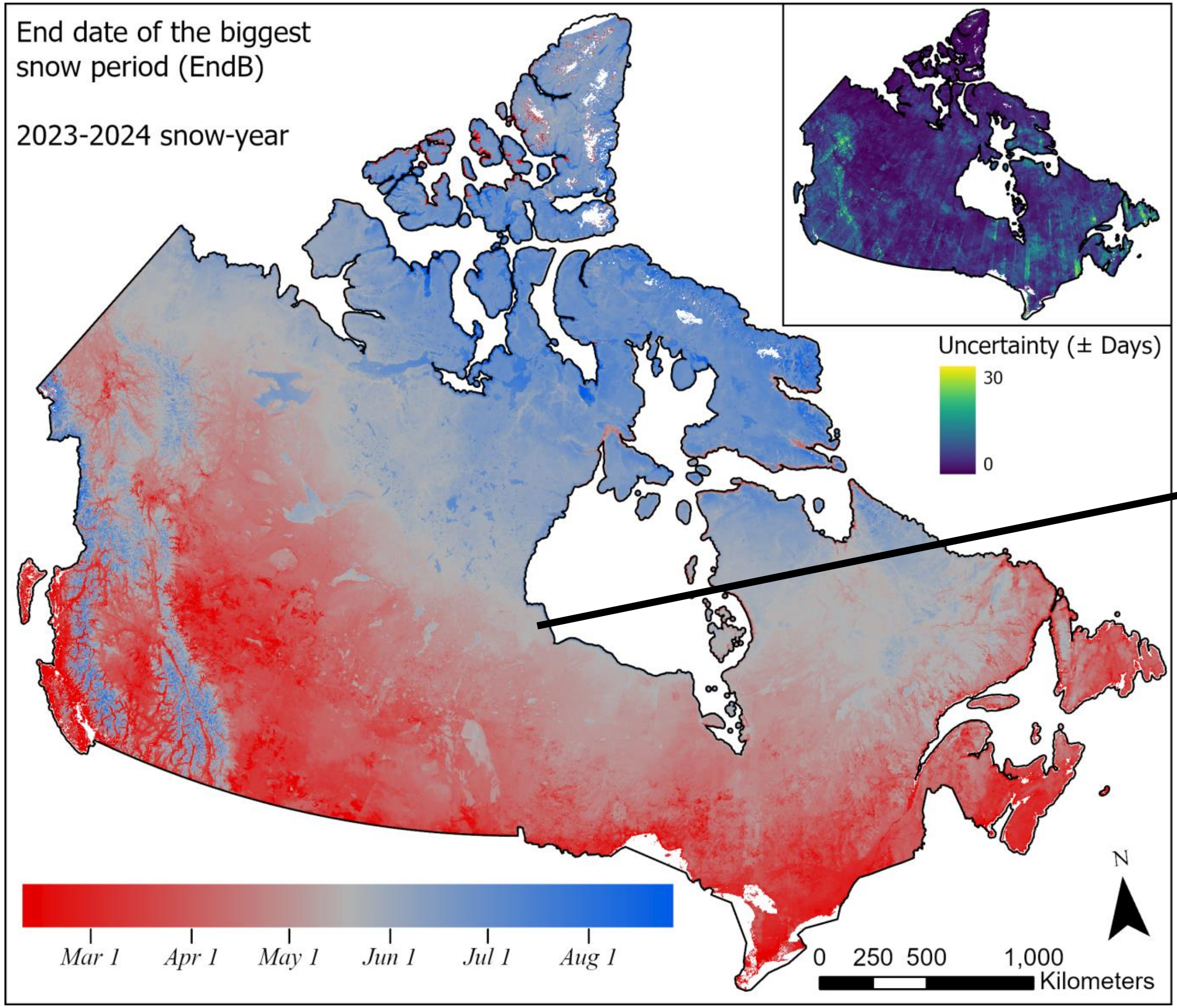


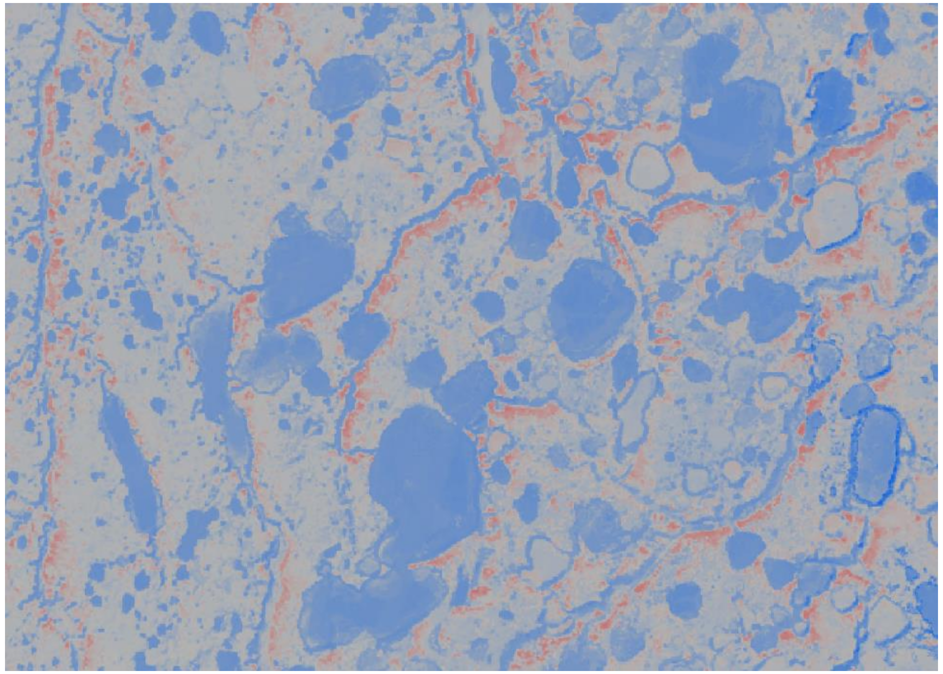


Example: Snow periods and status



Canada-wide Example





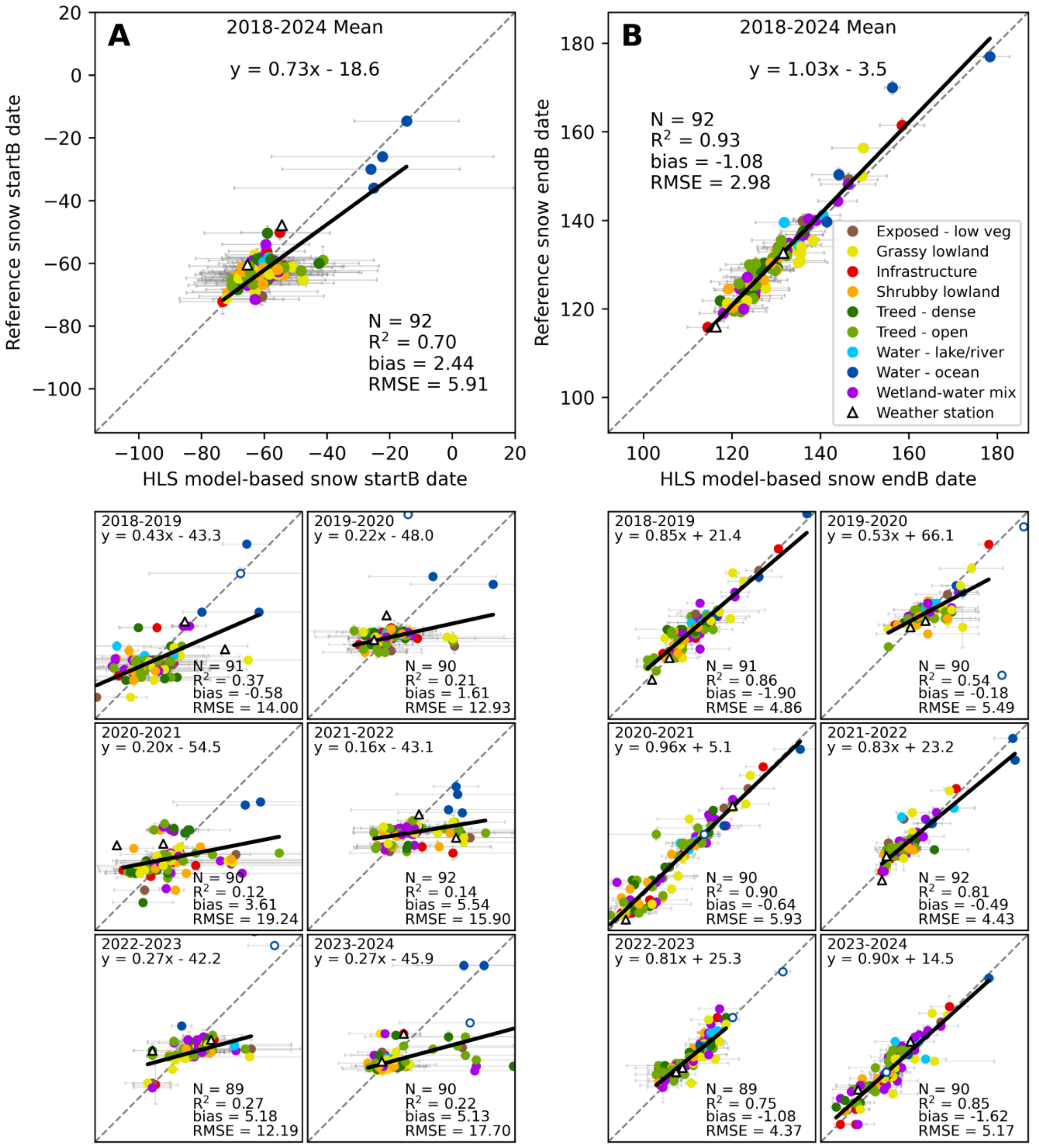
Zoomed into Wapusk area
93.2888222°W 57.5129447°N

Validation

- Interpretation of high-resolution imagery time-series: **In Progress**
  - Arctic Cordillera: 0/171
  - Atlantic Maritime: 0/166
  - Boreal Cordillera: 0/241
  - Boreal Plains: 0/357
  - Boreal Shield: 0/701
  - Hudson Plains: 92/230
    - Six snow-years for 92 samples
  - Mixedwood Plains: 0/134
  - Montane Cordillera: 0/277
  - Northern Arctic: 0/678
  - Pacific Maritime: 0/168
  - Prairies: 0/274
  - Southern Arctic: 0/456
  - Taiga Cordillera: 0/184
  - Taiga Shield: 0/644

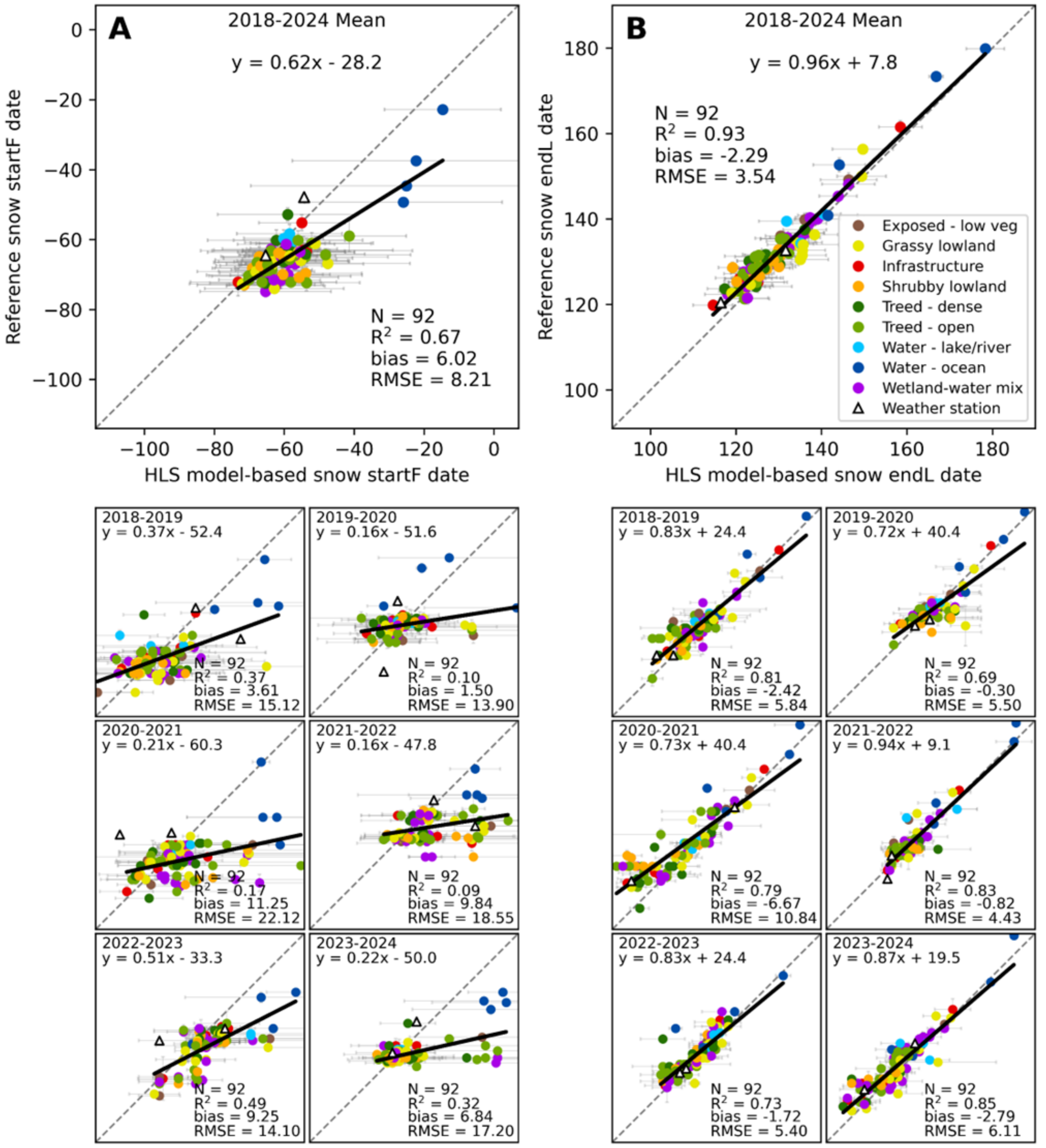
Initial Hudson Plains validation

Observed (imagery time-series interpretation reference) vs. predicted (HLS model-based) plots, multi-yearly (2018-2024, top) and for each snow-year (2018-2019 to 2023-2024, bottom). (A) Snow startB. (B) Snow endB. Snow-year plots have the same ranges as multi-year plots. Sample points (N = 92) are coloured by assessment group (Fig. A16). Sample points include error bars, with the x-direction quantifying model uncertainty and the y-direction quantifying interpretation uncertainty (Fig. A17). Data from two weather stations are shown but not included in statistics. Hollow-blue points are water – ocean samples where sea ice dynamics led to mid-winter non-snow interpretations for the given snow-year (e.g., short-duration polynyas). These points were also not included in statistics for biggest-period metrics, which can be noted by reduced sample sizes in some snow-years. Several of these points fall outside the plotting area. For startB: 2019-2020 (x = -71, y = 31), 2020-2021 (-18,37; -9,29), 2022-2023 (-11,22; -18,28), 2023-2024 (-2,52). For endB: 2019-2020 (92,77), 2020-2021 (175,76), 2022-2023 (96,79), 2023-2024 (180,26). See Figs. A33 and A34 for other metrics.



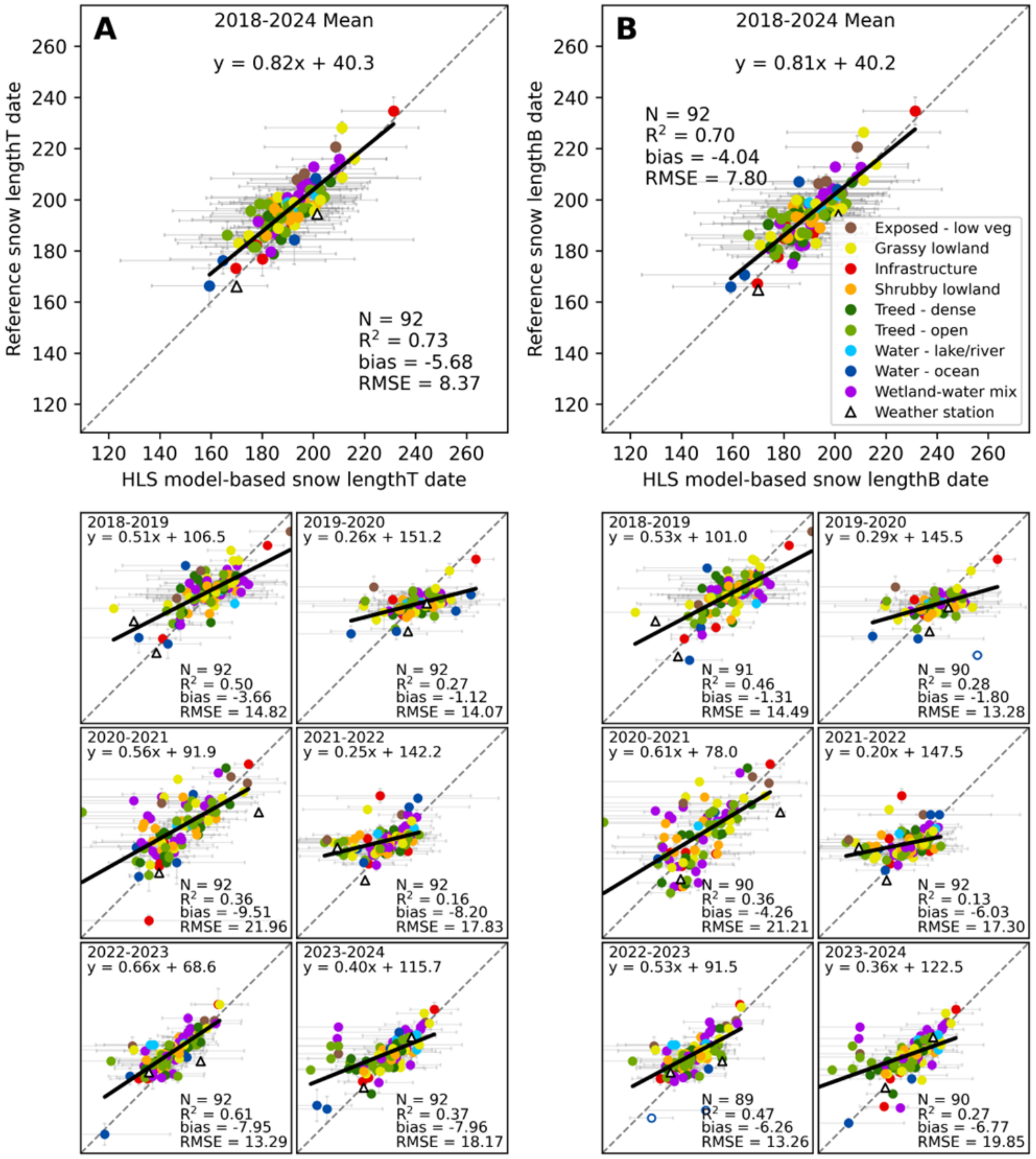
Initial Hudson Plains validation

Observed (imagery time-series interpretation reference) vs. predicted (HLS model-based) plots, multi-yearly (2018-2024, top) and for each snow-year (2018-2019 to 2023-2024, bottom). (A) Snow startF. (B) Snow endL. Format matches previous, except there are no hollow-blue points because water – ocean samples with mid-winter non-snow interpretations do not impact startF or endL.



Initial Hudson Plains validation

Observed (imagery time-series interpretation reference) vs. predicted (HLS model-based) plots, multi-yearly (2018-2024, top) and for each snow-year (2018-2019 to 2023-2024, bottom). (A) Snow lengthT. (B) Snow lengthB. Format matches previous, except there are no hollow-blue points for lengthT because water – ocean samples with mid-winter non-snow interpretations do not impact it much. Several sample points for lengthB fall outside the plotting area: 2018-2019 (113,98), 2019-2020 (x = 247, y = 82), 2020-2021 (155,100; 185,48), 2022-2023 (108,58), 2023-2024 (125,73; 190,64; 98,184).



Other validation

- Daily snow-depth from weather stations
  - Four stations in Hudson Plains analyzed
- Trail cameras: TBD
- Coincident very high-resolution validation: TBD

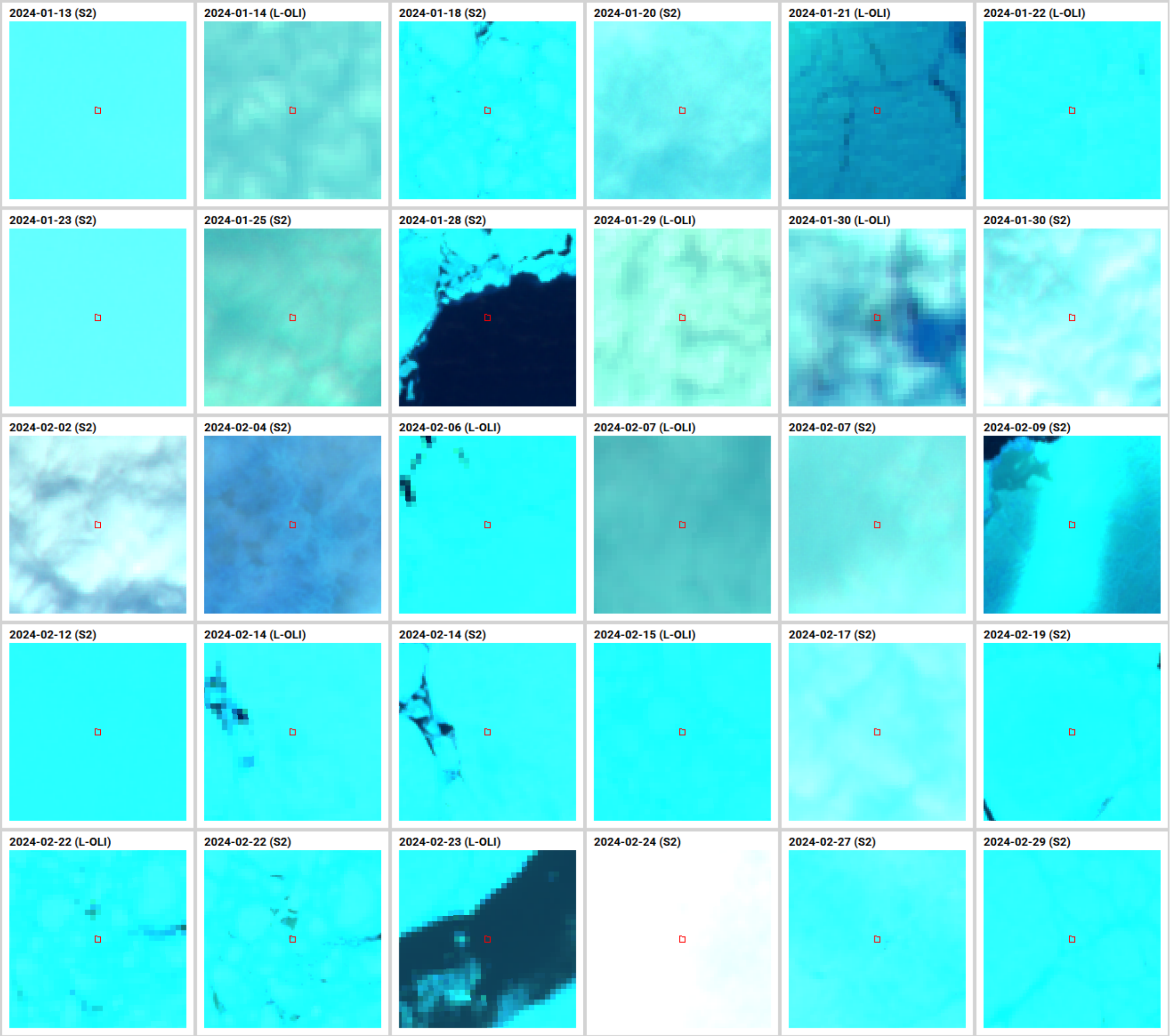
Issues to consider

- Model prone to missing short-duration (e.g., less than two weeks) snow or non-snow periods
  - Most impactful for areas with shifting sea ice dynamics, minor impacts on metrics that consider short-duration snow events (i.e., startF, endL, lengthT)
- Bright surfaces (e.g., beaches) can be confused for snow
  - Implausible snow removal helps
- Dense vegetation canopy may influence snow dynamics
  - Not sure scale of issue, or if issue at all (more validation required)
- Steep terrain can create shading issues
  - Most impactful for snow <u>start</u> and <u>length</u> in mountainous settings, generally does not impact snow <u>end</u> because melt usually happens in summer with better sun angles
  - Long winter darkness period at high latitudes can make this worse
- Landsat and Sentinel-2 orbit properties impact clear observation availability
  - Will see vertical slices with less observations than thin out towards the poles
  - No observations available for northern tip of Ellesmere Island

Short-duration snow/non-snow periods

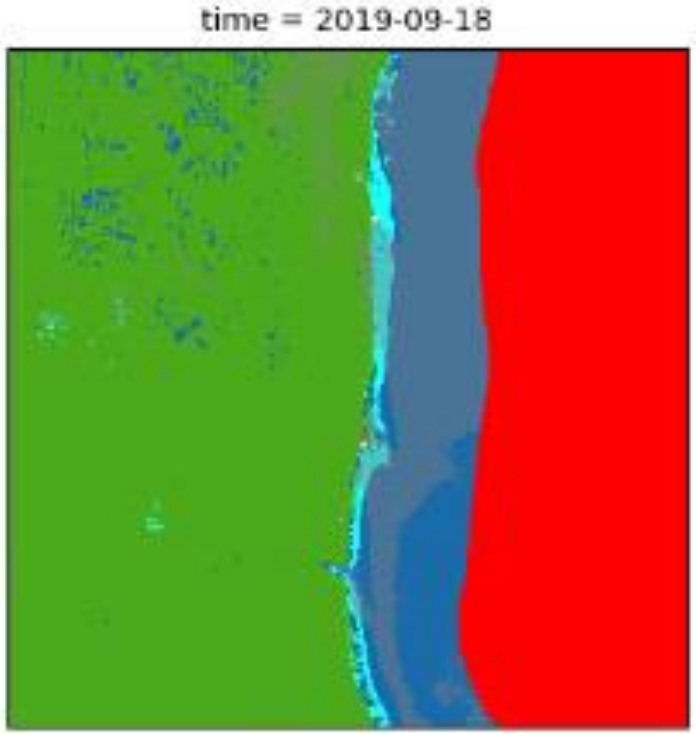
Model prone to missing short-duration (e.g., less than two weeks) snow or non-snow periods

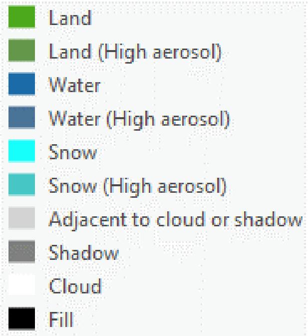
- Example shows sea ice in Hudson Bay between January and February 2024 visualized by Landsat and Sentinel-2
    - Pixel of interest is ice-free around January 28 and February 23, but these were not captured by the model – leading to large differences between observed and predicted endB dates (endL dates were similar).
- Short-duration snow events (e.g., snow falling after primary snow melt in spring and melting quickly after) are also often missed – leading to some difference between observed and predicted endL (and startF) dates.



Fmask Issue: Beaches, other turbid water

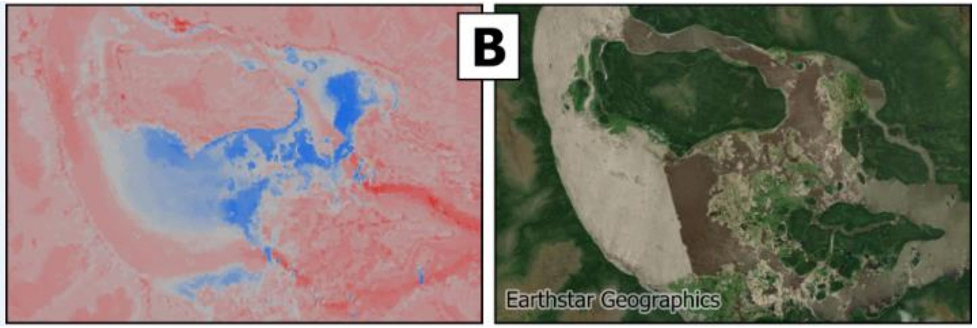
- Sometimes bright beaches/shallow water show up as snow in summer
  - IMS / uncertainty helps clean
  - Limitation of fmask algorithm for snow classification





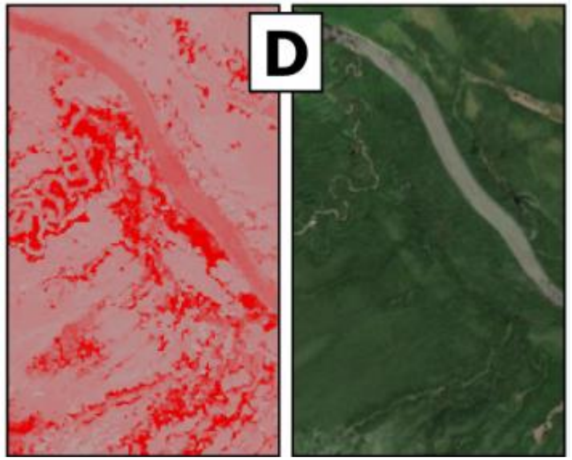
Red = Outside ROI

River rapids and sandy areas with later snow end dates.

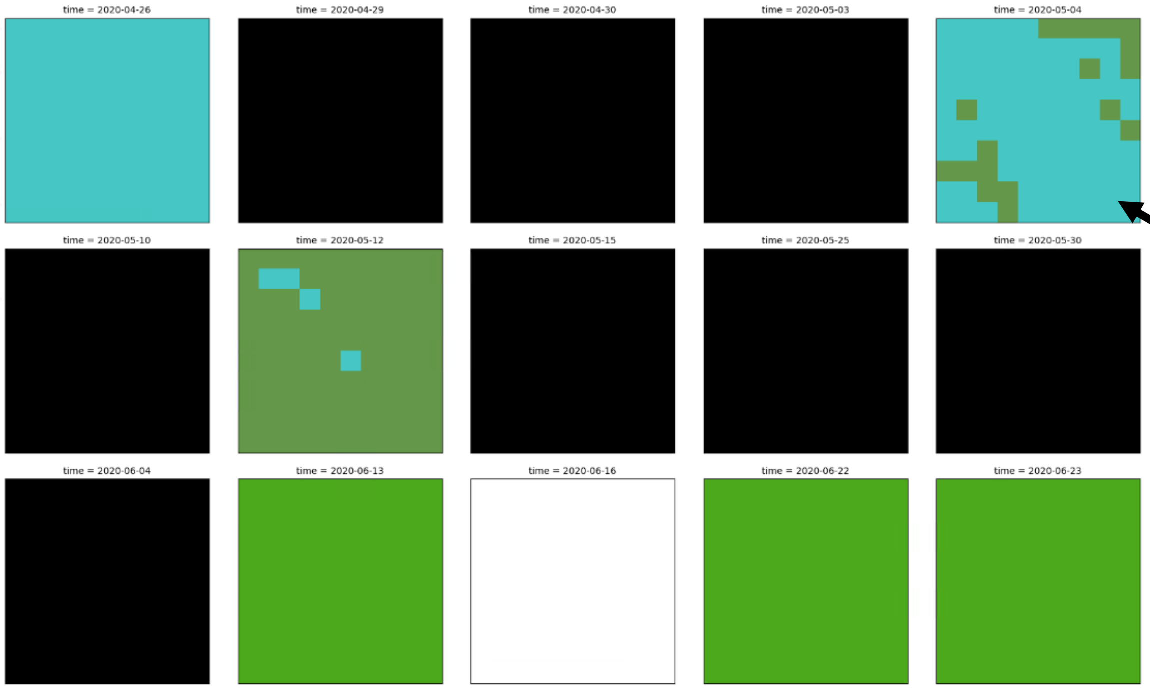


Inaccurate snow dynamics under forest canopy?

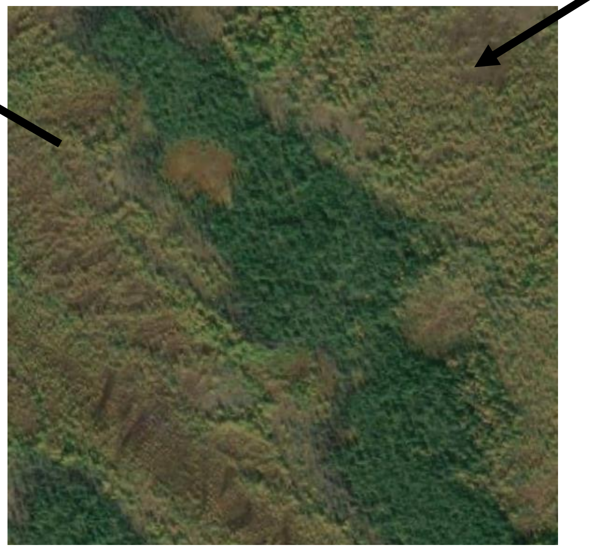
- In some cases, have seen later snow melt in dense forest – in other cases, earlier snow melt:



Early snow melt found in dense forest areas.



10 x 10 pixels around dense forest stand



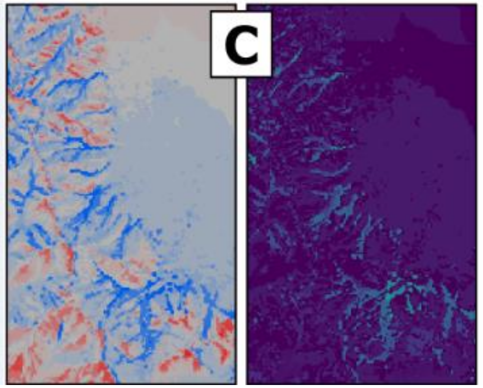
Approx area

- Open areas melt first (By May 4)
- Most rest (besides a few forested pixels) melt next (By May 12)
- Last few forest pixels melt last (By June 13 – Unlucky with lack of observations between May 12 and then)

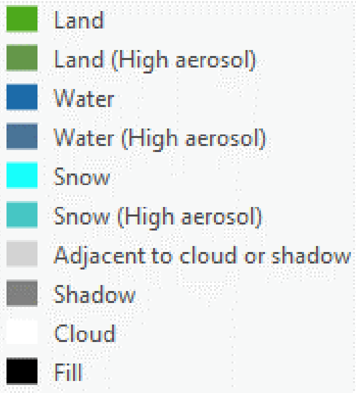
Steep terrain can create shading issues

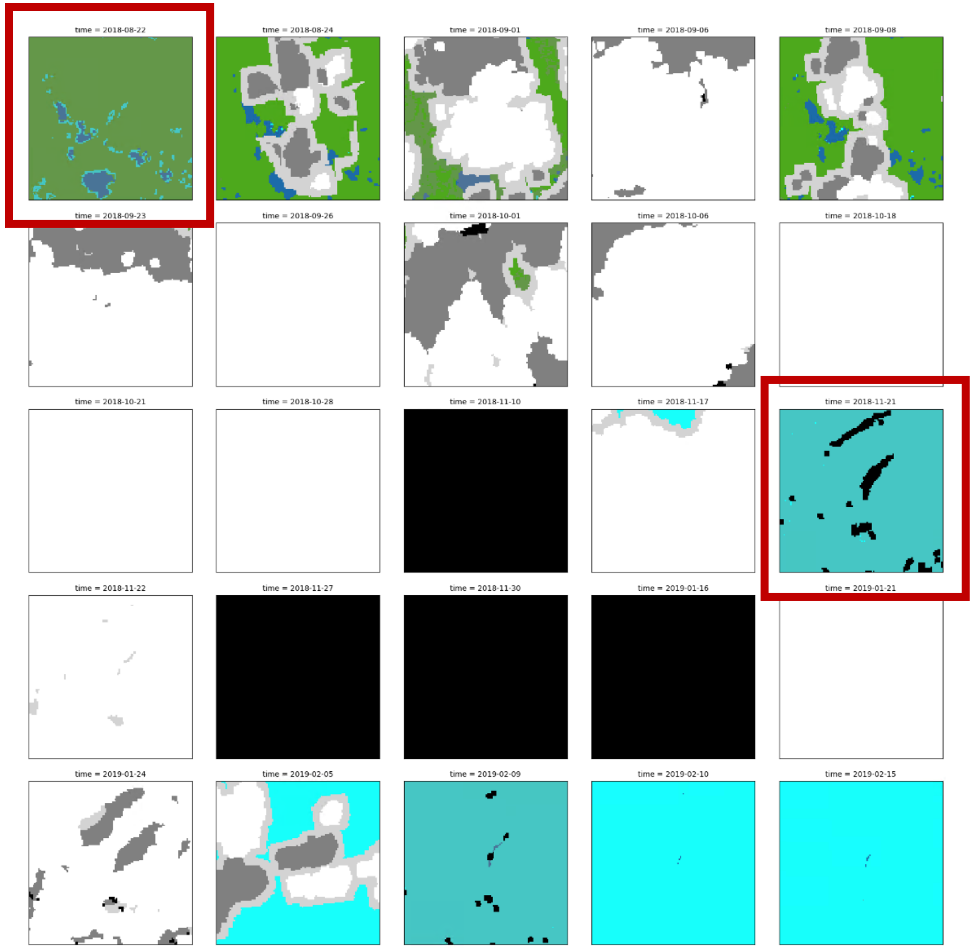
- This protected area between two steep hills had very late snow fall dates (Feb) because no clear observations between August and February

Some rugged terrain gets higher uncertainties too (right)



Some water body edges nearby had too early snow fall dates (Aug) because of multiple summer misclassifications in a row



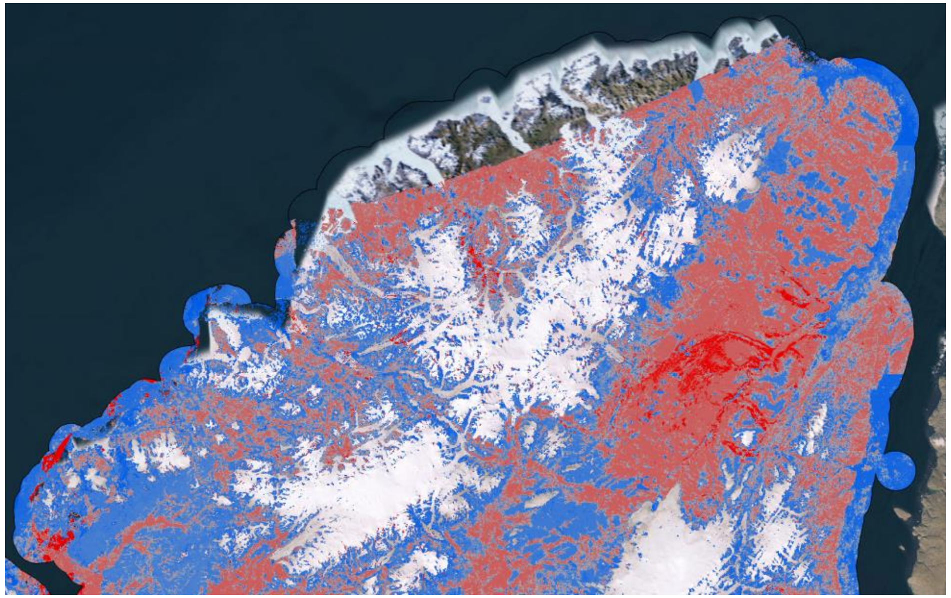


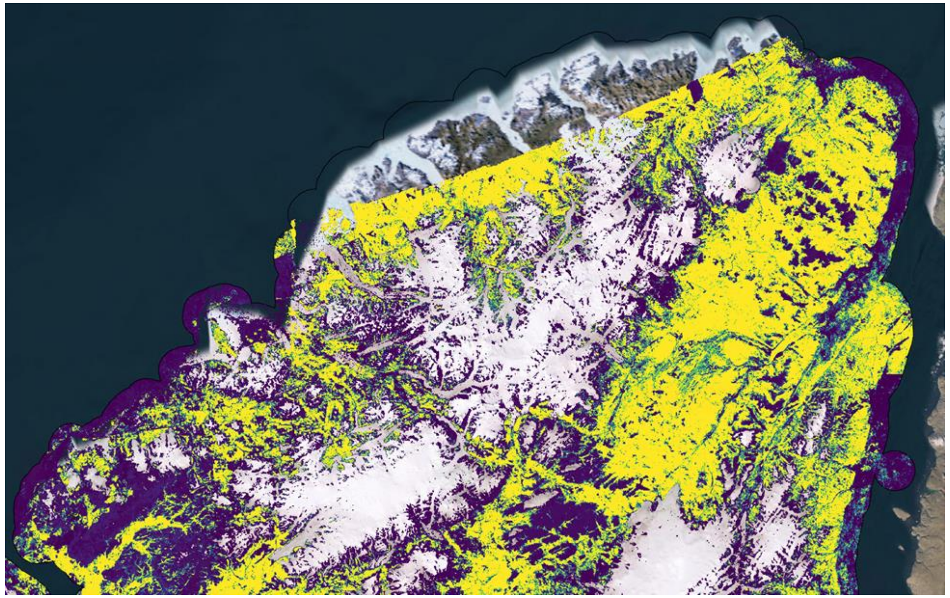
Long winter darkness period

- Impact on snow start, northern Ellesmere Island

Long winter darkness period means there is only a brief amount of time (if any) between first snow fall and when it is too dark for optical satellites to observe. This is not a problem for snow end date because snow melt happens when sun is high in the sky (e.g., June). Uncertainty is high in these scenarios (so can be used to filter) This issue becomes less pronounced as latitude decreases and depends on chance of fall snow observations in certain years.

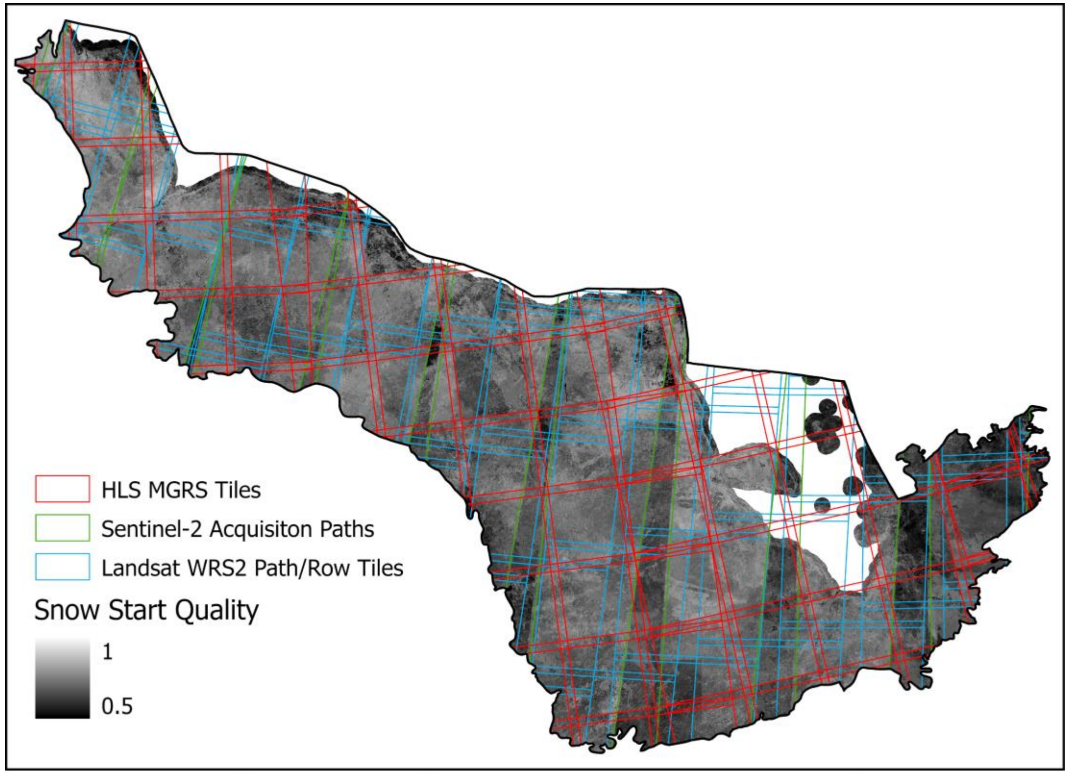
Also note lack of observations at northern tip of island, where no Sentinel-2 or Landsat data are collected.



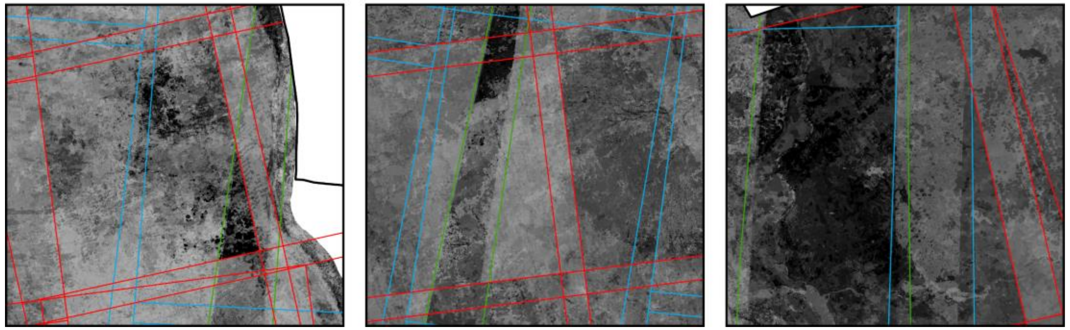


Landsat and Sentinel-2 orbit properties

- Due to the polar orbit structure of Landsat and Sentinel-2, observation frequency differs depending on orbit overlap
  - In Canada, as you move towards the North Pole, orbit overlap increases
  - In the right image, dark vertical slices represent areas with no orbit overlap and thus less clear observations.
    - Green in background is Sentinel-2's orbit, which corresponds to the wider slice



Note how variations in snow start quality occur along HLS processing tiles, and Sentinel-2/Landsat acquisition paths >



References

- Bauer-Marschallinger, B., Falkner, K., 2023. Wasting petabytes: A survey of the Sentinel-2 UTM tiling grid and its spatial overhead. *ISPRS Journal of Photogrammetry and Remote Sensing* 202, 682-690. https://doi.org/10.1016/j.isprsjprs.2023.07.015