Supplementary Material for

Remote sensing of chlorophyll-a in clear vs. turbid waters in large lakes

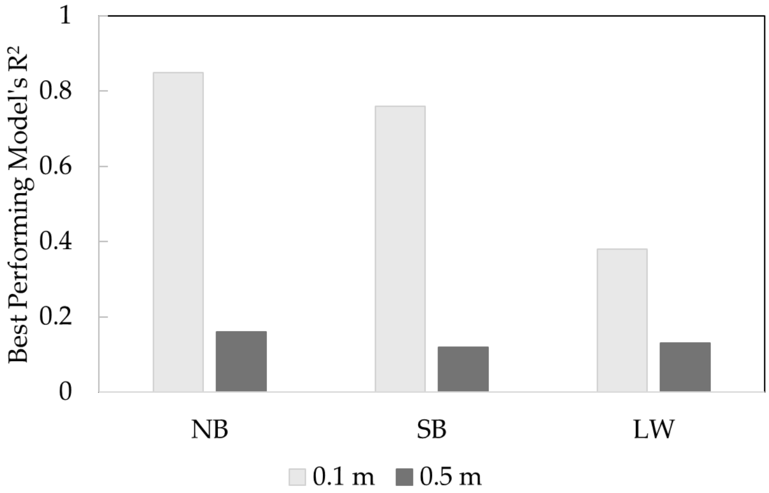

Figure S1. The R^2 of the best performing models using in situ matchups from different depths are displayed as light grey bars for 0.1 m and dark grey bars for 0.5 m. There is a five-fold reduction in R^2 when the depth of in situ matchups is increased from 0.1 to 0.5 m.

Table S1. The best calibration results for each tested Chl-*a* prediction model. The values indicate the highest R^2 for each Chl-*a* retrieval model among the 12 sets of matchups for different temporal (0 to ±3 days) and spatial windows (1×1, 3×3, and 5×5 pixels) for NB-specific and SB-specific, and LW-specific matchups. Any calibrations with no significant correlation between in situ and modeled chl-*a* ($p > 0.05$) are presented as a dash.

Band math	NB-Specific	SB-Specific	LW-Specific
B[1]	-	-	-
G[2]	0.24	-	-
R[3]	-	0.26	-
NIR	-	0.21	-
B/G[4]	0.82	-	0.33
B/R[6=5]	0.24	0.45	-
B/NIR	-	0.24	-
G/B[6]	0.85	-	0.38
G/R[7]	-	0.75	-
G/NIR	-	0.20	0.25
R/B[8]	0.29	0.42	-
R/G[9]	-	0.76	-
R/NIR[10]	-	-	-
NIR/B[11]	-	0.25	-
NIR/G	-	0.22	0.24
NIR/R[12]	-	-	-
B×G	-	-	-
B×R	-	-	-
B×NIR	-	-	-
G×R	0.18	-	-
G×NIR	-	-	-
R×NIR	-	0.26	-
Avg(R;NIR)	-	0.25	-
B×R×NIR	-	0.20	-
Ave(G;R)	0.19	-	-
Ave(B;R)	-	-	-
(R/B)×NIR	-	0.33	-
(G/R)×NIR	-	-	-
(R/G)×NIR	-	0.32	-
(B×G×R)	-	-	-
(B×G×NIR)	-	-	-

Band math	NB-Specific	SB-Specific	LW-Specific
(G×R×NIR)	-	0.22	-
Ave(B;G)	-	-	-
Ave(B;NIR)	-	-	-
Ave(G;NIR)	-	-	-
(B/R)×NIR	-	-	-
((1/B)-(1/G))×NIR	0.82	-	0.29
((1/R)-(1/G))×NIR	-	0.74	0.27
((1/R)-(1/B))×NIR	0.26	0.43	-
NDVI [13]	-	-	-
NRVI[14]	-	-	-
(B-R)×G	0.44	0.45	-
SABI[15]	-	-	-
KIVU [16]	0.43	0.44	-
Kab1[17]	-	-	0.37

(1) Ritchie et al (1990); (2) Lathrop & Lillesand (1986); Östlund et al (2001); (3) Tyler et al (2006); Allan et al (2011); Allan et al (2015); (4) Sudheer et al (2006); (5) Han & Jordan (2005); (6) Doxaran et al (2009); (7) Hellweger et al (2004); (8) Sass et al (2007); (9) Floricioiu et al (2004); (10) Strmbech et al (2004); (11) Tebbs et al (2013); (12) Duan et al (2007); (13) Normalized Difference Vegetation Index (NDVI) = ((NIR-R)/(NIR+R)) in Mishra & Mishra (2012); (14) Normalized Ratio Vegetation Index (NRVI) = (((R/NIR)-1)/((R/NIR)+1)) in Zhengjun et al (2008); (15) Surface Algal Bloom Index (SABI) = (NIR-R)/(B+G) in Alawadi (2010); (16) 3BDA-like (KIVU) = (B-R)/G in Brivio et al (2001); (17) Kab1 = 1.67-3.94×ln(B)+3.78×ln(G) in Kabbara et al (2008).

Table S2. The calibration and validation results for the NB-specific Best Performing Models (BPMs) for different spatial and temporal windows. The values are an average of the 5 cross-validation folds. Any calibrations with no significant correlation between in situ and modeled chl-*a* ($p > 0.05$) are presented as a dash.

		Model	**Calibration R^2**	**RMSE ($\mu g\ L^{-1}$)**	**RMSLE ($\mu g\ L^{-1}$)**	**NRMSE**	**MAE (μgL^{-1})**	**MAPE (%)**	**Bias**
1 × 1	0 days	-	-	-	-	-	-	-	-
	±1 day	G/B	0.85	31.94	0.74	1.98	22.81	67.72	9.39
	±2 days	$(B^{-1}-G^{-1})\times NIR$	0.75	20.86	0.61	0.73	13.57	50.44	2.04
	±3 days	$(B^{-1}-G^{-1})\times NIR$	0.57	23.37	0.72	0.88	14.48	38.68	7.25
3 × 3	0 days	-	-	-	-	-	-	-	-
	±1 day	G/B	0.70	46.11	0.99	3.11	32.66	115.39	9.81
	±2 days	G/B	0.66	23.93	0.72	0.71	16.49	43.11	0.37
	±3 days	$(B^{-1}-G^{-1})\times NIR$	0.61	33.41	0.75	1.01	21.72	34.21	-3.89
5 × 5	0 days	-	-	-	-	-	-	-	-
	±1 day	G/B	0.60	67.06	1.26	6.66	48.02	139.09	-5.86
	±2 days	$(B^{-1}-G^{-1})\times NIR$	0.58	39.81	0.78	0.95	26.06	65.33	-8.69
	±3 days	$(B^{-1}-G^{-1})\times NIR$	0.55	34.30	0.78	1.02	19.10	36.12	-1.35

Table S3. The calibration and validation results for the SB-specific Best Performing Models (BPMs) for different spatial and temporal windows. The values are average of the 5 cross-validation folds. Any calibrations with no significant correlation between in situ and modeled chl-*a* ($p > 0.05$) are presented as a dash.

		Model	**Calibration R^2**	**RMSE ($\mu g\ L^{-1}$)**	**RMSLE ($\mu g\ L^{-1}$)**	**NRMSE**	**MAE ($\mu g L^{-1}$)**	**MAPE (%)**	**Bias**
1 × 1	0 days	-	-	-	-	-	-	-	-
	±1 day	R/G	0.76	0.99	0.17	0.23	0.91	20.01	0.25
	±2 days	R/G	0.62	1.14	0.20	0.24	1	22.81	0.23
	±3 days	$(R^{-1}-G^{-1})\times NIR$	0.32	-	-	-	-	-	-
3 × 3	0 days	-	-	-	-	-	-	-	-
	±1 day	$(R^{-1}-G^{-1})\times NIR$	0.74	0.59	0.10	0.12	0.59	13.56	0.44
	±2 days	R/G	0.51	1.22	0.21	0.27	1.02	18.55	-0.08
	±3 days	-	-	-	-	-	-	-	-
5 × 5	0 days	-	-	-	-	-	-	-	-
	±1 day	R/G	0.63	2.35	0.37	0.35	2.35	54.30	2.35
	±2 days	R/G	0.63	1.20	0.20	0.26	1.02	19.75	1
	±3 days	-	-	-	-	-	-	-	-

Table S4. The calibration and validation results for the LW-specific Best Performing Models (BPMs) for different spatial and temporal windows. The values are average of the 5 cross validation folds. Calibrations with no significant correlation between in situ and modeled Chl-*a* ($p > 0.05$) were shown as dash.

		Model	**Calibration R^2**	**RMSE ($\mu g\ L^{-1}$)**	**RMSLE ($\mu g\ L^{-1}$)**	**NRMSE**	**MAE (μgL^{-1})**	**MAPE (%)**	**Bias**
1 × 1	0 days	-	-	-	-	-	-	-	-
	±1 day	G/B	0.38	-	-	-	-	-	-
	±2 days	NIR/G	0.17	-	-	-	-	-	-
	±3 days	-	-	-	-	-	-	-	-
3 × 3	0 days	-	-	-	-	-	-	-	-
	±1 day	G/B	0.35	-	-	-	-	-	
	±2 days	G/R	0.21	-	-	-	-	-	
	±3 days	G/B	0.09	-	-	-	-	-	-
5 × 5	0 days	-	-	-	-	-	-	-	-
	±1 day	G/B	0.27	-	-	-	-	-	-
	±2 days	G/R	0.29	-	-	-	-	-	-
	±3 days	-	-	-	-	-	-	-	-

References

1. Ritchie, J.C.; Cooper, C.M.; Schiebe, F.R. The relationship of MSS and TM digital data with suspended sediments, chlorophyll, and temperature in Moon Lake, Mississippi. *Remote Sens. Environ.* **1990**, *33*(2), 137-148. doi: https://doi.org/10.1016/0034-4257(90)90039-O.
2. Lathrop, R.G.; Lillesand, T.M. Use of Thematic Mapper data to assess water quality in Green Bay and central Lake Michigan. *PE&RS*, **1986**, 52(5), 671-680.
3. Östlund, C.; Flink, P.; Strömbeck, N.; Pierson, D.; Lindell, T. Mapping of the water quality of Lake Erken, Sweden, from imaging spectrometry and Landsat Thematic Mapper. *Sci. Total Environ.,* **2001**, 268(1-3), 139-154. doi: https://doi.org/10.1016/S0048-9697(00)00683-5.
4. Tyler, A.N.; Svab, E.; Preston, T.; Présing, M.; Kovács, W.A. Remote sensing of the water quality of shallow lakes: A mixture modelling approach to quantifying phytoplankton in water characterized by high-suspended sediment. *Int. J. of Remote Sens.* **2006**, 27(8), 1521-1537. doi: https://doi.org/10.1080/01431160500419311.
5. Allan, M.G.; Hamilton, D.P.; Hicks, B.J.; Brabyn, L. 2011. Landsat remote sensing of chlorophyll a concentrations in central North Island lakes of New Zealand. *Int. J. Remote Sens.* **2011**, 32(7), 2037-2055. doi: https://doi.org/10.1080/01431161003645840.
6. Allan, M.G.; Hamilton, D.P.; Hicks, B.; Brabyn, L. Empirical and semi-analytical chlorophyll a algorithms for multi-temporal monitoring of New Zealand lakes using Landsat. *Environ. Monit. Assess.* **2015**, 187, 1-24. doi: https://doi.org/10.1007/s10661-015-4585-4.
7. Sudheer, K.P.; Chaubey, I.; Garg, V. Lake water quality assessment from landsat thematic mapper data using neural network: an approach to optimal band combination selection1. *J. Am. Water Resour. Assoc.* **2006**, 42(6), 1683-1695. doi: https://doi.org/10.1111/j.1752-1688.2006.tb06029.x.
8. Han, L.; Jordan, K.J. Estimating and mapping chlorophyll-a concentration in Pensacola Bay, Florida using Landsat ETM+ data. *Int. J. Remote Sens.* **2005**, 26(23), 5245-5254. doi: https://doi.org/10.1080/01431160500219182.
9. Doxaran, D.; Froidefond, J.M.; Castaing, P.; Babin, M. Dynamics of the turbidity maximum zone in a macrotidal estuary (the Gironde, France): Observations from field and MODIS satellite data. *Estuar. Coast. Shelf Sci.* **2009**, 81(3), 321-332. doi: https://doi.org/10.1016/j.ecss.2008.11.013.
10. Hellweger, F.L.; Miller, W.; Oshodi, K.S. Mapping turbidity in the Charles River, Boston using a high-resolution satellite. *Environ. Monit. Assess.* **2007**, 132, 311-320. doi: https://doi.org/10.1007/s10661-006-9535-8.
11. Sass, G.Z.; Creed, I.F.; Bayley, S.E.; Devito, K.J. Understanding variation in trophic status of lakes on the Boreal Plain: A 20-year retrospective using Landsat TM imagery. *Remote Sens. Environ.* **2007**, 109(2), 127–141, doi: 10.1016/j.rse.2006.12.010.
12. Floricioiu, D.; Rott, H.; Rott, E.; Dokulil, M.; Defrancesco, C. 2003. Retrieval of limnological parameters of perialpine lakes by means of MERIS data. In Proceedings of

the **2004** Envisat & ERS Symposium (ESA SP-572), 6–10 September 2004, Salzberg, Austria, pp. 1–5 (Paris: ESA).

13. Strömbeck, N.; Candiani, G.; Giardino, C.; Zilioli, E. Water quality monitoring of Lake Garda using multi-temporal MERIS data. In Proceedings of MERIS User Workshop (ESA SP-549), 10–13 November **2003**, Frascati, Italy, p. 17.1 (Paris: ESA).
14. Tebbs, E.J.; Remedios, J.J.; Harper, D.M. (2013). Remote sensing of chlorophyll-a as a measure of cyanobacterial biomass in Lake Bogoria, a hypertrophic, saline- alkaline, flamingo lake, using Landsat ETM. *Remote Sens. Environ*. **2013**, 135, 92-106. doi:10.1016/j.rse.2013.03.024.
15. Duan, H.; Zhang, Y.; Zhang, B.; Song, K.; Wang, Z. Assessment of chlorophyll-a concentration and trophic state for Lake Chagan using Landsat TM and field spectral data. *Environ. Monit. Assess*. **2007**, 129(1), 295- 308. doi:10.1007/s10661-006-9362-y
16. Mishra, S.; Mishra, D.R. Normalized difference chlorophyll index: A novel model for remote estimation of chlorophyll-a concentration in turbid productive waters. *Remote Sens Environ* **2012,** 117, 394-406. doi: 10.1016/j.rse.2011.10.016.
17. Zhengjun, W.; Jianming, H.; Guisen, D. Use of satellite imagery to assess the trophic state of Miyun Reservoir, Beijing, China. *Environ. Pollut*. **2008**, 155(1), 13-19. doi: https://doi.org/10.1016/j.envpol.2007.11.003.
18. Alawadi, F. Detection of surface algal blooms using the newly developed algorithm surface algal bloom index (SABI). In Remote Sensing of the Ocean, Sea Ice, and Large Water Regions **2010** (Vol. 7825, p. 782506). International Society for Optics and Photonics. doi: https://doi.org/10.1117/12.862096.
19. Brivio, P.A.; Giardino, C.; Zilioli, E. 2001. Determination of chlorophyll concentration changes in Lake Garda using an image-based radiative transfer code for Landsat TM images. *Int. J. Remote Sens.* **2001**, 22(2-3), 487-502. doi: https://doi.org/10.1080/014311601450059.
20. Kabbara, N.; Benkhelil, J.; Awad, M.; Barale, V. Monitoring water quality in the coastal area of Tripoli (Lebanon) using high-resolution satellite data. *ISPRS J. Photogramm. Remote Sens.* **2008**, 63(5), 488-495. doi: https://doi.org/10.1016/j.isprsjprs.2008.01.004.

```
// =============================================
// datasets.js
// =============================================
// This script is part of the series of scripts for Landsat atmospheric correction & matchups
extraction. It calls Landsat dataset.
// ------------------------------------------------------------------
// Author: Forough Fendereski and Ian Mackenzie
// ------------------------------------------------------------------


exports.landsat5 = {
  collection: ee.ImageCollection("LANDSAT/LT05/C02/T1"),
  bands: {
    blue: { number: 1, eSun: 1958, dP: 0.02874, tR: 0.162672, tOz: 0.0087 },
    green: { number: 2, eSun: 1827, dP: 0.02825, tR: 0.084703, tOz: 0.0309 },
    red: { number: 3, eSun: 1551, dP: 0.02792, tR: 0.046362, tOz: 0.0167 },
    nir: { number: 4, eSun: 1036, dP: 0.02755, tR: 0.017491, tOz: 0.0002 },
    swir1: { number: 5, eSun: 214.9, dP: 0.02724, tR: 0.00109, tOz: 0.0063 },
    qa: { name: "QA_PIXEL", allowed: [672, 676] },
  },
};

exports.landsat7 = {
  collection: ee.ImageCollection("LANDSAT/LE07/C02/T1"),
  bands: {
    blue: { number: 1, eSun: 1970, dP: 0.02874, tR: 0.165452, tOz: 0.0087 },
    green: { number: 2, eSun: 1842, dP: 0.02825, tR: 0.090387, tOz: 0.0309 },
    red: { number: 3, eSun: 1547, dP: 0.02792, tR: 0.045797, tOz: 0.0167 },
    nir: { number: 4, eSun: 1044, dP: 0.02755, tR: 0.017918, tOz: 0.0002 },
    swir1: { number: 5, eSun: 225.7, dP: 0.02724, tR: 0.001167, tOz: 0.0063 },
    qa: { name: "QA_PIXEL", allowed: [672, 676] },
  },
};

exports.landsat8 = {
  collection: ee.ImageCollection('LANDSAT/LC08/C02/T1'),
  bands: {
    blue: { number: 2, eSun: 2067, dP: 0.02874, tR: 0.165452, tOz: 0.0087 },
    green: { number: 3, eSun: 1893, dP: 0.02825, tR: 0.088441, tOz: 0.0309 },
    red: { number: 4, eSun: 1603, dP: 0.02792, tR: 0.047814, tOz: 0.0167 },
    nir: { number: 5, eSun: 972.6, dP: 0.02755, tR: 0.015541, tOz: 0.0002 },
    swir1: { number: 6, eSun: 245, dP: 0.02724, tR: 0.001281, tOz: 0.0063 },
    qa: { name: "QA_PIXEL", allowed: [2720, 2724] },
  },
};

exports.windSpeed = {
  collection: ee.ImageCollection("ECMWF/ERA5_LAND/HOURLY"),
  bands: {
    u: "u_component_of_wind_10m",
    v: "v_component_of_wind_10m",
   mld: "lake_mix_layer_depth",
   tp: "total_precipitation",
   tph: "total_precipitation_hourly",
   rh: "runoff_hourly",
   srh: "surface_runoff_hourly",
  },
};

exports.all = [exports.landsat5, exports.landsat7, exports.landsat8, exports.windSpeed];
```

```
// ==================================================
// preprocessing.js
// ==================================================
// This script is part of the series of scripts for Landsat atmospheric correction & matchups
extraction. It masks out pixels affected by cloud and cloud shadow (only keeps pixels that are
water) and performs atmospheric correction on Landsat Level 1 data.
// ------------------------------------------------------------------
// Author: Forough Fendereski and Ian Mackenzie
// ------------------------------------------------------------------


function bandName(bandNumber) {
  return "B" + bandNumber.toString();
}

function calibratedReflectanceBand(image, bandProperties) {
  var bandNumber = bandProperties.number;
  var original = image.select(bandName(bandNumber));
  var gain = image.metadata("RADIANCE_MULT_BAND_" + bandNumber.toString());
  var bias = image.metadata("RADIANCE_ADD_BAND_" + bandNumber.toString());
  var toaRadiance = original.multiply(gain).add(bias);

//   var pi = ee.Image.constant(Math.PI);
//   var degToRad = ee.Image.constant(Math.PI / 180.0);
//   var sunElevationDeg = image.metadata("SUN_ELEVATION");
//   var sunZenithDeg = ee.Image.constant(90.0).subtract(sunElevationDeg);
//   var sunZenithRad = sunZenithDeg.multiply(degToRad);
//   var scatterDeg = ee.Image.constant(180.0).subtract(sunZenithDeg);
//   var scatterRad = scatterDeg.multiply(degToRad);

//   var yVar = ee.Image.constant(bandProperties.dP / (2.0 - bandProperties.dP));
//   var cosScatter = scatterRad.cos();
//   var pRay = toaRadiance.expression(
//     "0.75 * ((1.0 - yVar) / (1.0 + 2.0 * yVar)) * (1.0 + cosScatter ** 2) + ((3.0 * yVar) / (1.0 + 2.0 *
yVar))",
//     { yVar: yVar, cosScatter: cosScatter }
//   );

//   var negativeTOz = ee.Image.constant(-bandProperties.tOz);
//   var tOzUp = negativeTOz.exp();
//   var cosSunZenith = sunZenithRad.cos();
//   var tOzDown = negativeTOz.divide(cosSunZenith).exp();
//   var tR = ee.Image.constant(bandProperties.tR);
//   var one = ee.Image.constant(1.0);
//   var negativeOne = ee.Image.constant(-1.0);
//   var rRayExp = one
//     .divide(cosSunZenith)
//     .add(one)
//     .multiply(tR)
//     .multiply(negativeOne)
//     .exp(); // exp(-1 * tR * ((1 / (cos(sunZenithRad))) + 1))
//   var eSun = ee.Image.constant(bandProperties.eSun);
//   var rayleighPathRadiance = pRay.expression(
//     "((eSun * cosSunZenith * pRay) / (4.0 * pi * (cosSunZenith + 1.0))) * (1.0 - rRayExp) * tOzUp *
tOzDown",
//     {
//       pRay: pRay,
//       eSun: eSun,
//       cosSunZenith: cosSunZenith,
//       pi: pi,
//       one: one,
//       rRayExp: rRayExp,
//       tOzUp: tOzUp,
//       tOzDown: tOzDown,
//     }
//   );

// var atmCorrToaRadiance = toaRadiance.subtract(rayleighPathRadiance);

  var sunDistance = image.metadata("EARTH_SUN_DISTANCE");
  return toaRadiance
    .multiply(sunDistance)
    .multiply(sunDistance)
    .multiply(pi)
    .divide(eSun.multiply(cosSunZenith));
 }

function fourBands(image, bands) {
  return image.select(
    [
      bandName(bands.red.number),
      bandName(bands.green.number),
      bandName(bands.blue.number),
      bandName(bands.nir.number),
    ],
    ["RED", "GREEN", "BLUE", "NIR"]
  );
}

function applyQaMask(image, bands) {
  var DILATED_CLOUD = 1 << 0;
  var CLOUD = 1 << 3;
  var CLOUD_SHADOW = 1 << 4;
  var WATER = 1 << 7;
  var bitmask = ee.Image.constant(WATER).uint16();

  var qaBand = image.select(bands.qa.name);
  var qaMask = qaBand.bitwiseAnd(bitmask).neq(ee.Image(0));

  return image.updateMask(qaMask);
}

function calibratedReflectance(image, bands) {
  var maskedImage = applyQaMask(image, bands);
  return fourBands(
    ee.Image([
      calibratedReflectanceBand(maskedImage, bands.blue),
      calibratedReflectanceBand(maskedImage, bands.green),
      calibratedReflectanceBand(maskedImage, bands.red),
      calibratedReflectanceBand(maskedImage, bands.nir),
    ]),
    bands
  );
}

exports.fourBands = fourBands;
exports.calibratedReflectance = calibratedReflectance;
```

```
// ==================================================
// sandbox.js
// ==================================================
// This script is part of the series of scripts for Landsat atmospheric correction & matchups
extraction. It masks out pixels affected by cloud and cloud shadow (only keeps pixels that are
water) and performs atmospheric correction on Landsat Level 1 data.
// ---------------------------------------------------------------------
// Author: Forough Fendereski and Ian Mackenzie
// ---------------------------------------------------------------------


var datasets = require("users/foroughfendereski/Forough:datasets.js");
var preprocessing = require("users/foroughfendereski/Forough:preprocessing.js");

// function filteredCollection(collection) {
//   var startDate = ee.Date("1999-10-01");
//   var endDate = ee.Date("1999-11-01");

//   return collection
//     .filterDate(startDate, endDate)
//     .filterBounds(coastline)
//     .filter(
//       ee.Filter.and(
//         ee.Filter.gte("SUN_ELEVATION", 30.0),
//         ee.Filter.lte("SUN_ELEVATION", 60.0),
//         ee.Filter.lte("CLOUD_COVER", 90.0)
//       )
//     );
// }

function exportImage(image, fileName) {
  Export.image.toDrive({
    image: image,
    description: fileName,
    scale: 30,
    fileFormat: 'GeoTIFF',
    maxPixels: 1e13
  });
}

// LT05_L1TP_031025_19930617_20170118_01_T1 (old)
// LT05_L1TP_031025_19930617_20200914_02_T1 (new)

// LE07_L1TP_030024_20020729_20170130_01_T1 (old)
// LE07_L1TP_030024_20020729_20200916_02_T1 (new)

// LC08_L1TP_031025_20130912_20170309_01_T1 (old)
// LC08_L1TP_031025_20130912_20200912_02_T1 (new)

var image5 =
datasets.landsat5.collection.filter(ee.Filter.eq("LANDSAT_PRODUCT_ID","LT05_L1TP_031025_
19930617_20200914_02_T1")).first()
var image7 =
datasets.landsat7.collection.filter(ee.Filter.eq("LANDSAT_PRODUCT_ID","LE07_L1TP_030024_
20020729_20200916_02_T1")).first();
var image8 = datasets.landsat8.collection.filter(ee.Filter.eq("LANDSAT_PRODUCT_ID",
"LC08_L1TP_031025_20130912_20200912_02_T1")).first();

print(image5);
print(image7);
print(image8);

function exportPair(image, bands, landsat) {
  var stripped = preprocessing.fourBands(image, bands);
  var processed = preprocessing.calibratedReflectance(image, bands, coastline);
  exportImage(stripped, "gee-preprocessing-landsat-" + landsat + "-original");
  exportImage(processed, "gee-preprocessing-landsat-" + landsat + "-processed");
}

exportPair(image5, datasets.landsat5.bands, "5");
exportPair(image7, datasets.landsat7.bands, "7");
exportPair(image8, datasets.landsat8.bands, "8");

// var bands5 = datasets.landsat5.bands;
// var processed5 = preprocessing.calibratedReflectance(image5, datasets.landsat5.bands);
// print(processed5);

// exportImage(preprocessing.fourBands(image5, datasets.landsat5.bands), "gee-preprocessing-
test-original-2");
// exportImage(processed5, "gee-preprocessing-test-processed-2");
// ==================================================
// export-matchup-data.js
// ==================================================
// This script is part of the series of scripts for Landsat atmospheric correction & matchups
extraction. It takes data points containing time and location of Chl-a samples and extracts
Landsat matchups within the specified spatial and temporal windows.
// ----------------------------------------------------------------------
// Author: Forough Fendereski and Ian Mackenzie
// ----------------------------------------------------------------------



var datasets = require("users/foroughfendereski/Forough:datasets.js");
var preprocessing = require("users/foroughfendereski/Forough:preprocessing.js");

function estimatedWindImage(date, pointCollection, prefix) {
  return datasets.windSpeed.collection
    //.filterDate(date.advance(-40, "minute"), date.advance(40, "minute"))
    .filterDate(date.advance(-90, "minute"), date.advance(90, "minute"))
    .filterBounds(pointCollection)
    .mean()
    //.select([datasets.windSpeed.bands.u, datasets.windSpeed.bands.v], [prefix + "WindEast",
prefix + "WindNorth"]);
    .select(
     [
      datasets.windSpeed.bands.u,
      datasets.windSpeed.bands.v,
      datasets.windSpeed.bands.mld,
      datasets.windSpeed.bands.tp,
      datasets.windSpeed.bands.tph,
      datasets.windSpeed.bands.rh,
      datasets.windSpeed.bands.srh
     ],
     [
      prefix + "WindEast",
      prefix + "WindNorth",
      prefix + "MLD",
      prefix + "TotPrecipitation",
      prefix + "TotPrecipitationHourly",
      prefix + "RunoffHourly",
      prefix + "SurfaceRunoffHourly"
  ]
  );
}

function pointSamplesFromDataset(point, landsatDataset) {
 var sampleDate = ee.Date(point.get("SAMPLING_D"), "America/Winnipeg");
 var landsatStartDate = sampleDate.advance(-5, "day");
 var landsatEndDate = sampleDate.advance(5, "day");
 var pointCollection = ee.FeatureCollection([point]);

 var landsatImages = landsatDataset.collection
  .filterDate(landsatStartDate, landsatEndDate)
  .filterBounds(pointCollection);

 var sampleWindImage = estimatedWindImage(sampleDate, pointCollection, "sample");

 function sampleDataAtPoint(landsatImage) {
  var reflectance = preprocessing.calibratedReflectance(landsatImage, landsatDataset.bands);

  function rename(image, suffix) {
   return image.select(
    ["BLUE", "GREEN", "RED", "NIR","SWIR1"],
    ["blue" + suffix, "green" + suffix, "red" + suffix, "nir" + suffix, "swir1" + suffix]
   );
  }

  var reflectance1x1 = rename(reflectance, "1x1");
  var reflectance3x3 = rename(reflectance.focalMedian(1.5, "square"), "3x3");
  var reflectance5x5 = rename(reflectance.focalMedian(2.5, "square"), "5x5");
  var landsatWindImage = estimatedWindImage(landsatImage.date(), pointCollection,
"landsat");
  var combinedImage = ee.Image([reflectance1x1, reflectance3x3, reflectance5x5,
sampleWindImage, landsatWindImage]);

  function addProperties(feature) {
   return feature.set({
    landsat_product_id: landsatImage.get("LANDSAT_PRODUCT_ID"),
    cloud_cover: landsatImage.get("CLOUD_COVER"),
    sun_azimuth: landsatImage.get("SUN_AZIMUTH"),
    sun_elevation: landsatImage.get("SUN_ELEVATION"),
    difference: landsatImage.date().difference(sampleDate, "day"),
        lt_overpas: landsatImage.date().format(),
      });
    }

    return combinedImage.sampleRegions(pointCollection).map(addProperties).first();
  }

  return landsatImages.map(sampleDataAtPoint, true);
}

function pointSamples(point) {
  return pointSamplesFromDataset(point, datasets.landsat5)
    .merge(pointSamplesFromDataset(point, datasets.landsat7))
    .merge(pointSamplesFromDataset(point, datasets.landsat8));
}

var results = inputs.map(pointSamples).flatten();

Export.table.toDrive(results, "matchups-4GEE-RefWindMLDPreciRunoff_swir");

//print(results);
```

```
// =================================================
// export-modelled-chl-images.js
// =================================================
// This script applies the Chl-a prediction models on pre-processed Landsat data and export
Chl-a images.
// -----------------------------------------------------------------
// Author: Forough Fendereski and Ian Mackenzie
// -----------------------------------------------------------------



var datasets = require("users/foroughfendereski/Forough:datasets.js");
var preprocessing = require("users/foroughfendereski/Forough:preprocessing.js");

var timeZone = "America/Winnipeg";

function reflectanceImages(dataset, basin, startDate, endDate) {
  function reflectanceImage(image) {
    return preprocessing.calibratedReflectance(image, dataset.bands);
  }
  return dataset.collection.filterDate(startDate,
endDate).filterBounds(basin).map(reflectanceImage);
}

function basinReflectanceImages(basin, startDate, endDate) {
  var images5 = reflectanceImages(datasets.landsat5, basin, startDate, endDate);
  var images7 = reflectanceImages(datasets.landsat7, basin, startDate, endDate);
  var images8 = reflectanceImages(datasets.landsat8, basin, startDate, endDate);
  return images5.merge(images7).merge(images8);
}

function chlModel(image, numeratorBand, denominatorBand, a, b) {
  var numerator = image.select(numeratorBand);
  var denominator = image.select(denominatorBand);
  return
numerator.divide(denominator).multiply(ee.Image(a)).add(ee.Image(b)).exp().rename("MODE
LLED_CHL");
}

function northBasinModel(image) {
  // ln Chl-a = 8.1527 × [G/B] - 5.6221
  return chlModel(image, "GREEN", "BLUE", 8.1527, -5.6221);
}

function southBasinModel(image) {
  // ln Chl-a = 4.3126 × [R/G] - 2.6028
  return chlModel(image, "RED", "GREEN", 4.3126, -2.6028)
}

function exportMeanMaxImages(basin, model, label, startDate, endDate) {
  var modelledChlImages = basinReflectanceImages(basin, startDate, endDate).map(model);
  Export.image.toDrive({
    image: modelledChlImages.max(),
    description: label + "_max",
    region: basin,
    scale: 30
  });
  Export.image.toDrive({
    image: modelledChlImages.mean(),
    description: label + "_mean",
    region: basin,
    scale: 30
  });
}

function exportImages(basin, model, label, year) {
  for (var month = 7; month <= 10; month++) {
    var start = ee.Date.fromYMD(year, month, 1, timeZone);
    var mid = ee.Date.fromYMD(year, month, 16, timeZone);
    var end = ee.Date.fromYMD(year, month + 1, 1, timeZone);
    var monthString = month.toString();
    if (month < 10) {
      monthString = "0" + monthString;
    }
    var monthLabel = label + "_" + year + "_" + monthString;
    exportMeanMaxImages(basin, model, monthLabel + "a", start, mid);
    exportMeanMaxImages(basin, model, monthLabel + "b", mid, end);
  }
}

exportImages(northBasin, northBasinModel, "NB", 2011);
```

```
// var startDate = ee.Date.fromYMD(year, 7, 1, timeZone);
// var endDate = ee.Date.fromYMD(year, 10, 31, timeZone);
// print(datasets.landsat5.collection.filterDate(startDate,
endDate).filterBounds(northBasin).aggregate_array("DATE_ACQUIRED").sort());


s.first()), ls_viz, 'Landsat');
```