DuRoss, C.B., Briggs, R.W., Gold, R.D., Hatem, A.E., Elliott, A.J., Delano, J., Medina-Cascales, I., Gray, H.J., Mahan, S.A., Nicovich, S., Lifton, Z.M., Kleber, E., McDonald, G., Hiscock, A., Bunds, M., and Reitman, N.G., 2021, How similar was the 1983 M_w 6.9 Borah Peak earthquake rupture to its surface-faulting predecessors along the northern Lost River fault zone (Idaho, USA)?: GSA Bulletin, <https://doi.org/10.1130/B36144.1>.

Supplemental Material

Text S1. Bayesian (OxCal) models for northern Lost River fault zone trench sites

Text S2. Bulk sediment analysis and charcoal identification

Text S3. Luminescence geochronology

Table S1. Description of stratigraphic units at the Sheep Creek trench

Table S2. Description of stratigraphic units at the Arentson Gulch trench

Figure S1. Photomosaics and large-format trench logs for the Sheep Creek trench

Figure S2. Photomosaics and large-format trench logs for the Arentson Gulch trench

Figure S3. Sheep Creek and Arentson Gulch vertical displacement measurements

Figure S4. Fault bend angles along the northern Lost River fault zone

Figure S5. Photographs of the Sheep Creek and Arentson Gulch trench sites

Figure S6. Probability density functions for Lost River fault zone ruptures

How similar was the 1983 M_w 6.9 Borah Peak earthquake rupture to its surface-faulting predecessors along the northern Lost River fault zone (Idaho, USA)?

Christopher B. DuRoss¹, Richard W. Briggs¹, Ryan D. Gold¹, Alexandra E. Hatem¹, Austin J. Elliott², Jaime Delano¹, Ivan Medina-Cascales³, Harrison J. Gray⁴, Shannon A. Mahan⁴, Sylvia Nicovich⁵, Zachery M. Lifton⁶, Emily Kleber⁷, Greg McDonald⁷, Adam Hiscock⁷, Mike Bunds⁸, and Nadine G. Reitman¹

¹ U.S. Geological Survey, Golden, CO

² U.S. Geological Survey, Moffett Field, CA

³ University of Alicante, Spain

⁴ U.S. Geological Survey, Denver, CO

⁵ Bureau of Reclamation, Denver, CO

⁶ Idaho Geological Survey, Boise, ID

⁷ Utah Geological Survey, Salt Lake City, UT

⁸ Utah Valley University, Orem, UT

Text S1.

Supporting information for this manuscript includes Bayesian models constructed using (1) previously published stratigraphic and geochronologic data from the Rattlesnake Canyon, 25-km, Poison Spring, Doublespring Pass, Elkhorn Creek, Lone Cottonwood Canyon, and Lower Cottonwood Canyon trench sites (Hait and Scott, 1978; Cochran, 1985; Scott et al., 1985; Schwartz and Crone, 1985, 1988; Vincent, 1995; Haller and Crone, 2004), and (2) new paleoseismic data (this study) for the Sheep Creek and Arentson Gulch trench sites. All models were constructed using OxCal version 4.4 (Bronk Ramsey, 2009).

Rattlesnake Canyon

Plot()

```
{
Sequence("RC_v1; Dec1, 2020")
{
Boundary("sequence start");
C_Date("postglacial fan",-12980,1000);
Boundary("early Holocene fan sedimentation");
Date("RC1");
R_Date("SC88_1", 4949, 200);
Boundary("wedge sedimentation");
Boundary("historical record",1900 AD,50);
};
};
```

25-km Trench

Plot()

```
{
Sequence("25KM_v1; Dec1, 2020")
{
Boundary("sequence start");
C_Date("postglacial fan",-12980,1000);
Boundary("early Holocene fan sedimentation");
Date("25KM1");
R_Date("SC88_2", 5280, 240);
Boundary("wedge sedimentation");
Boundary("historical record",1900 AD,50);
};
};
```

Sheep Creek (this study)

Plot()

```
{
Sequence("SC_v1; Nov16, 2020")
{
Boundary("Sequence start");
//Unit 1
C_Date("postglacial fan",-12980,1000);
//R_Date("SC-R13", 3230, 110); [burrowed?]
Date("SC2");
Phase("Unit C2")
{
R_Date("SC-R9a", 8800, 40);
R_Date("SC-R9b", 8670, 40);
R_Date("SC-R4b", 8560, 80);
R_Date("SC-R9c", 7940, 30);
R_Date("SC-R5b", 6870, 30);
R_Date("SC-R5a", 6150, 30);
//R_Date("SC-R6a", 4320, 25);
//R_Date("SC-R6b", 3890, 20);
//R_Date("SC-R6c", 5230, 25);
//R_Date("SC-R11a", 5470, 25);
//R_Date("SC-11b") modern
//R_Date("SC-R11c", 4910, 35);
};
//Phase("Unit 1 under C1")
//{
//R_Date("SC-R15a", 6990, 30);
//R_Date("SC-R18b", 6810, 30);
```

```

//R_Date("SC-R18a", 6650, 30);
//R_Date("SC-R18c", 5580, 30);
//R_Date("SC-R15b", 5310, 20);
//};
Date("SC1");
Phase("Unit C1")
{
  //Delta_R("broaden uncertainty",0,250);
  R_Date("SC-R8b", 5740, 25);
  R_Date("SC-R8a", 5610, 30);
  R_Date("SC-R7c", 5470, 20);
  R_Date("SC-R7b", 5440, 25);
  R_Date("SC-R16", 4690, 25);
  R_Date("SC-R17b", 4120, 25);
  //R_Date("SC-R7a", 7750, 25);
  //R_Date("SC-R22a", 8700, 40);
  //R_Date("SC-R22b", 8750, 75);
  //R_Date("SC-R22c", 8640, 35);
};
Phase("Unit 3 and uppermost C1")
{
  R_Date("SC-R19b", 4110, 20);
  R_Date("SC-R19a", 3180, 20);
  R_Date("SC-R17a", 2740, 20);
};
//R_Date("SC-R3", 8070, 35); [Cx]
//R_Date("SC-R23a", 6190, 45); [Cx]
//R_Date("SC-R23b", 6390, 25); [Cx]
Boundary("historical record");
};
};

```

Arentson Gulch (this study)

```

Plot()
{
  Sequence("AG_v1; Nov16, 2020")
  {
    Boundary("sequence start");
    //Unit 1
    C_Date("postglacial fan",-12980,1000);
    Date("AG2");
    Phase("Unit C2")
    {
      R_Date("AG-R5", 8110, 70);
      R_Date("AG-R7", 6610, 30);
    }
  }
}

```



```

//R_Date("AG-R1", 4930, 90);
};
Date("AG1");
Phase("Unit C1")
{
//R_Date("AG-R10", 7260, 30);
R_Date("AG-R3", 5360, 140);
R_Date("AG-R11", 3910, 20);
};
Phase("Unit 5")
{
R_Date("AG-R4a", 1820, 40);
R_Date("AG-R4b", 2170, 20);
};
Boundary("historical record");
};
};

```

Poison Spring

```

Plot()
{
Sequence("PS_v2; Dec1, 2020")
{
Boundary("sequence start");
Date("PS2");
R_Date("V95", 9045, 100);
//Boundary("wedge sedimentation");
Date("PS1");
Boundary("wedge sedimentation");
Boundary("historical record",1900 AD,50);
};
};

```

Doublespring Pass

```

Plot()
{
Sequence("DP_v1; Dec14, 2020")
{
Boundary("sequence start");
C_Date("postglacial fan",-12980,1000);
Date("DP1");
//Boundary("wedge deposition");
C_Date("scarp age",-3980,1000);
Boundary("historical record",1900 AD,50);

```

```
};  
};
```

Elkhorn Creek

```
Plot()  
{  
  Sequence("EC_v1; Dec1, 2020")  
  {  
    Boundary("sequence start");  
    C_Date("postglacial fan",-12980,1000);  
    R_Date("V95", 9710, 240);  
    Date("EC1");  
    Boundary("fan deposition");  
    Boundary("historical record",1900 AD,50);  
  };  
};
```

Lone Cottonwood Canyon

```
Plot()  
{  
  Sequence("LCC_v1; Dec14, 2020")  
  {  
    Boundary("sequence start");  
    C_Date("postglacial fan",-12980,1000);  
    Date("LCC2");  
    C_Date("Mazama ash", -5677, 150);  
    //Mazama ash: Zdanowicz et al. (1999)  
    Date("LCC1");  
    Boundary("historical record",1900 AD,50);  
  };  
};
```

Lower Cottonwood Canyon

```
Plot()  
{  
  Sequence("LWC_v1; Dec14, 2020")  
  {  
    Boundary("sequence start");  
    C_Date("postglacial fan",-12980,1000);  
    Date("LWC2");  
    R_Date("Glacier Peak",11600,50);  
    //Glacier Peak: Kuehn et al. (2009)  
    C_Date("Mazama ash", -5677, 150);  
  };  
};
```

```

Date("LWC1");
R_Date("Scott1985",4320,130);
//need to verify age is 14C yr
Boundary("fan deposition");
Boundary("historical record",1900 AD,50);
};
};

```

REFERENCES CITED

- Bronk Ramsey, C., 2009, Bayesian analysis of radiocarbon dates: *Radiocarbon*, v. 51, p. 337–360, doi:10.1017/S0033822200033865.
- Cochran, B.D., 1985, Age of Late Quaternary surface ruptures along the Thousand Springs–Mackey segment of the Lost River Range fault system in Jacobson, M. L., and Rodriguez, T. R., eds., *National Earthquake Hazards Reduction Program, Summaries of technical reports, Volume XXI: U.S. Geological Survey Open-File Report 86-31*, p. 113–121.
- Hait, M.H., and Scott, W.E., 1978, Holocene faulting, Lost River Range, Idaho: *Geological Society of America Abstracts with Programs*, v. 10, p. 217.
- Haller, K.M., and Crone, A.J., 2004, Twenty years after the Borah Peak earthquake: Field guide to surface-faulting earthquakes along the Lost River fault, Idaho, in Haller, K.M., and Wood, S.H., eds., *Geological Field Trips in Southern Idaho, Eastern Oregon, and Northern Nevada: U.S. Geological Survey Open-File Report 2004-1222*, p. 118–135.
- Schwartz, D.P., and Crone, A.J., 1988, Paleoseismicity of the Lost River fault zone, Idaho: Earthquake recurrence and segmentation: *Geological Society of America Abstracts with Programs*, v. 20, p. 228.
- Schwartz, D.P., and Crone, A.J., 1985, The 1983 Borah Peak earthquake—A calibration event for quantifying earthquake recurrence and fault behavior on Great Basin normal faults, in Stein, R. S., and Bucknam, R. C., eds., *Proceedings of workshop XXVIII on the Borah Peak, Idaho, earthquake: U.S. Geological Survey Open-File Report 85-290*, v. A, p. 153–160.
- Scott, W.E., Pierce, K.L., and Hait, M.H., Jr., 1985, Quaternary tectonic setting of the 1983 Borah Peak earthquake, central Idaho: *Bulletin of the Seismological Society of America*, v. 75, p. 1053–1066.
- Vincent, K.R., 1995, Implications for models of fault behavior from earthquake surface-displacement along adjacent segments of the Lost River fault, Idaho: *University of Arizona*, 152 p.

Text S2.

BULK SEDIMENT ANALYSIS OF SAMPLES FROM ARENTSON GULCH AND
THE SHEEP CREEK TRENCH, LOST RIVER RANGE, CUSTER COUNTY, IDAHO

By

Jenna Battillo
and
Linda Scott Cummings

With assistance by
Caitlin A. Clark

PaleoResearch Institute
Golden, Colorado

PaleoResearch Institute Technical Report 2019-099

Prepared for

U.S. Geological Survey
Denver, Colorado

November 2019

INTRODUCTION

The Arentson Gulch and Sheep Creek sites are located in the Lost River Range of Custer County, Idaho, approximately 35 km southeast of Challis. Thirty-two bulk sediment samples (10 from Arentson Gulch and 22 from Sheep Creek), collected from two trenches, were sent to the PaleoResearch Institute for flotation and macrofloral analysis with the goal of isolating and identifying materials for radiocarbon dating.

METHODS

Bulk Sediment

Bulk sediment samples were floated using a modification of procedures outlined by Matthews (1979). Each sample was added to approximately three gallons of water, then stirred until a strong vortex formed. The floating material (light fraction) was poured through a 250-micron mesh sieve. All material that passed through the screen was retained for possible microcharcoal, particulate soil organics, and/or humate extraction. Additional water was added and the process repeated until all floating material was removed from the samples (a minimum of five times). The material remaining in the bottom (heavy fraction) was poured through a 0.5-mm mesh screen. The floated portions were allowed to dry.

The light fractions were weighed, then passed through a series of graduated screens (US Standard Sieves with 4-mm, 2-mm, 1-mm, 0.5-mm, and 0.25-mm openings) to separate charcoal debris and to initially sort the remains. Contents of each screen then were examined. Charcoal pieces larger than 2 mm, 1 mm, 0.5 mm in diameter were separated from the rest of the light fraction, and the total charcoal was weighed. When they were recovered, charcoal larger than 2 mm were weighed and recorded separately. Charcoal pieces in a representative sample were broken to expose fresh cross, radial, and tangential sections, then examined under a binocular microscope at a magnification of up to 75x and, if necessary, using a Nikon Optiphot 66 microscope at magnifications of 320–800x. Weights of each charcoal type within the representative sample were recorded. Material that remained in the 4-mm, 2-mm, 1-mm, and 0.5-mm sieves was scanned under a binocular stereo microscope at a magnification of 10x, with some identifications requiring magnifications of up to 75x. Material that passed through the 0.5-mm screen was quickly scanned for smaller inclusions such as sclerotia and insect remains. Heavy fractions were scanned at a magnification of 2x for the presence of botanic remains. Sample constituents that were not separated for weighing were noted with an X in the macrofloral table, followed by an approximation of their frequency within the sample: abundant, moderate, or few. For friable charcoals and other constituents, the number of fragments are estimated and noted with an asterisk (*) to indicate that the number is an approximate. The term "seed" is used to represent seeds, achenes, caryopses, and other disseminules. Remains from the light and heavy fractions were recorded as charred and/or uncharred, whole and/or fragments. Macrofloral remains, including charcoal, were identified using manuals (Carlquist 2001; Hoadley 1990; Martin and Barkley 1961; Musil 1963; Schopmeyer 1974; Schweingruber et al. 2011, 2013) and by comparison with modern and archaeological references. Clean laboratory conditions were used during flotation and identification to avoid contamination of charcoal and botanic remains to be submitted for

radiocarbon dating. All instruments were washed between samples, and the samples were protected from contact with modern charcoal.

DISCUSSION

Thirty-two bulk sediment samples from the Arentson Gulch and Sheep Creek sites, located in the Lost River Range of Custer County, Idaho, were floated and analyzed for macrofloral remains (Table 1) aimed toward isolating and identifying charred botanical remains suitable for radiocarbon dating. Small, rather than large, pieces of charcoal were common. In fact, much of the charcoal recovered from these sites was friable, readily breaking into smaller pieces when handled. Large pieces of charcoal are more resistant to degradation during chemical pretreatment, while friable charcoal loses more mass during processing. Therefore, samples of friable charcoal require more mass for radiocarbon dating than do single pieces of charcoal.

The Arentson Gulch Site

The Arentson Gulch site trench, located at approximately 2300 m above mean sea level (amsl), includes alluvial and colluvial sediments, as well as a buried A horizon, and zones of cemented calcareous sediments. The sediments are dominated by silts and gravels with sandy inclusions. Evidence for two surface faulting earthquakes (AG1 and AG2) is apparent in this trench. Vegetation at the trench location is dominated by *Artemisia* (sagebrush) and grasses (Tables 2 and 3), but conifers and other trees are abundant just to the east.

Ten bulk sediment samples from the Arentson Gulch site were analyzed for macrofloral remains. Two samples (both from Unit 2) contained no charcoal, while three others did not contain charcoal adequate for dating. Charcoal from this site was friable, yielding a heavily fragmented population that continued to break apart during handling. Furthermore, all but one of these samples contained less than 5 mg, which is considered by some labs as a borderline quantity that is unlikely to survive chemical treatment. However, these samples could be processed at PRI.

Unit 2

The samples from Unit 2 (AG-R6 and AG-R9) did not contain any charcoal. The only botanical remains recorded in these samples were all modern (uncharred), aside from some potentially charred resin fragments noted in Sample AG-R6, which is not recommended for radiocarbon dating.

Unit C2S

Samples AG-R5 and AG-R7 from Unit C2S both contained conifer charcoal. Sample AG-R5 yielded 2.8 mg of conifer charcoal and 1.2 mg of unidentifiable charcoal. We suggest combining these for AMS analysis for a total of 4 mg.

Sample AG-R7 did not contain enough charcoal to date.

Unit C2

Sample AG-R8 was the only sample examined from Unit C2. A very small, charred conifer needle and one small fragment of conifer charcoal were the only charred plant remains identified in this sample. This sample did not contain enough charcoal to date.

Unit 3

Sample AG-R1 was the only sample examined from Unit 3. It contained 1.5 mg of conifer charcoal, which is not an adequate amount for dating.

Unit C1

Two samples from Unit C1 were submitted: AG-R10 and AG-R11. Both of these samples contained sufficient charred plant material to date.

Sample AG-R10 contained a very small amount of charcoal (both conifer and unidentifiable). In addition, this sample contained 2 mg of charred parenchymous tissue. Generally, parenchymous tissue is not recommended for dating because it could be from roots, but it could also come from stems or other plant parts. This sample did not yield an adequate quantity of charcoal to date without combining with the parenchymous tissue, which is not recommended.

AG-R11 contained a total of 2.6 mg of charcoal, which included conifer, vitrified conifer, and unidentifiable charcoal, which should be combined to obtain a quantity sufficient for radiocarbon dating.

Unit 4

AG-R3 was the only sample examined from Unit 4. This sample contained three unidentified charred seed fragments (2 mg) and 4 fragments of unidentified charred plant tissue weighing 0.6 mg. The charcoal was primarily conifer (1.7 mg), but also included two fragments of unidentified hardwood (0.6 mg). We suggest combining the charcoal for AMS radiocarbon dating.

Unit 5

Only one sample was submitted from Unit 5: AG-R4. The total charcoal from this sample weighed 5.7 mg. This sample contained enough *Artemisia* charcoal to date (four fragments equaling 3.1 mg). If this quantity is still too small, the *Artemisia* charcoal could be

combined with the conifer charcoal (1.5 mg) and also the unidentified hardwood and unidentified charcoal to boost the sample size.

The Sheep Creek Site

The Sheep Creek trench sits at an elevation of approximately 2385 m amsl. This trench was excavated into a mixture of alluvial and colluvial sediments also dominated by silty gravel and gravely silt with some larger cobbles. Two surface faulting earthquakes also were apparent at this trench (SC1 and SC2). The vegetation at the trench also is dominated by sagebrush and grasses. Stands of conifers and deciduous trees occupy hillslopes to the east and are present in the drainage.

Twenty-two bulk sediment samples from the Sheep Creek Site were analyzed for macrofloral remains. Overall, the charcoal from these samples was dominated by conifer (Tables 3 and 4). Most of the charcoal was friable and too small to identify, but fragments from a few samples were identified as *Juniperus* (Juniper), *Pinus* (Pine), and as *Pinus*, 3-5 needle type (a subgroup of pines characterized by clusters of 3-5 needles that includes *Pinus ponderosa* (ponderosa pine), *Pinus monicola* (western white pine), and others. *Betula* (birch) charcoal, recovered only in Sample SC-R6, was the only hardwood charcoal identified, although several samples contained fragments of unidentified hardwood charcoal. One sample (SC-R3) contained two charred nutshell fragments identified as probable *Corylus* (hazel).

Several samples did not contain any charcoal (SC-R1, SC-R12, SC-R14, SC-R21, and SC-R24), or did not yield sufficient charcoal to date (SC-R2, SC-R13, and SC-R16). The majority of samples contained numerous very small fragments of charcoal that need to be combined to boost sample size for radiocarbon dating. In total, 14 samples contained sufficient charcoal to date. Seven of these 14 samples yielded 5 mg or less of datable charcoal, limiting selection of radiocarbon dating facility to obtain a radiocarbon age. PaleoResearch Institute, for instance, has a proven record in radiocarbon dating small samples of 5 mg or less. The only sample to yield a single piece of charcoal (juniper) large enough to date (0.0126 g) is SC-R22. It was broken in half for identification.

Unit 1

Unit 1, located in an alluvial fan sediment composed of gravely cobbles with silt, includes a localized sub-unit (Unit 1Bk) of cemented soil carbonate. A total of eight samples were examined from Unit 1, including one from Unit 1Bk. Of the seven samples from Unit 1, only two (SC-R15 and SC-R18) displayed sufficient charcoal to date. Sample SC-R15 contained a total of 4.7 mg of charcoal, including mostly conifer and some unidentifiable charcoal. Dating this sample may require combining conifer and unidentified charcoal, since the charcoal was friable and highly fragmentary.

Sample SC-R18 yielded more than enough charcoal to date. Five fragments of conifer charcoal weighing 6.4 mg were selected for dating, but these could be combined with the rest of the conifer charcoal (approximately 35 fragments weighing 7.5 mg), if desired.

Sample SC-R12, from Unit 1Bk (the cemented carbonate zone within Unit 1), did not contain any charcoal. This sample yielded only two small fragments of charred plant remains that were identified as possible seeds. Additional organic components of this sample comprised rootlets and faunal bone.

Unit 2

Unit 2 comprises alluvial sediment (gravely silt with cobbles) that caps Unit 1. Sample SC-R24 was the only sample examined from Unit 2. This sample did not contain any charcoal or charred plant remains.

Unit C2

Unit C2 is a silty gravel with cobbles that includes colluvium in the eastern fault zone. This unit postdates earthquake SC2 and is deposited atop Unit 1. Samples from this unit contained more charcoal and a greater variety of charcoal than those from Units 1 and 2. Charcoal from these samples included conifer, *Pinus* (3-5 needle type), *Betula* (birch), and unidentified hardwood. All five samples from this unit yielded sufficient charcoal for radiocarbon dating.

Sample SC-R4 contained a total of 5 mg of probable conifer charcoal, which should be adequate for dating. Other constituents of this sample faunal bone, PET, and other miscellaneous charred and uncharred botanical remains. The term PET (processed edible tissue) was originated by Nancy Stenholm (1994) and refers to softer tissue types, such as starchy parenchymoid or fruity epitheloid tissues. Although the term “PET” stems from archaeological investigations where plant processing is likely, it is applied to geological samples as representing charred starchy or fruity plant tissues that are incorporated into the sediment record from natural wildfires in a similar manner to incorporation of charcoal. PET fruity tissues resemble sugar-laden fruit or berry tissue without seeds, as well as tissue from succulent plant parts such as cactus pads.

Sample SC-R5 contained 2.2 mg conifer charcoal, less than 1 mg unidentified hardwood charcoal, and 0.7 mg unidentifiable charcoal. We suggest combining these charcoals to obtain a date. This sample also included faunal bone and a rodent tooth, as well as PET, rootlets, and other uncharred floral remains. Recovery of a rodent tooth suggests bioturbation.

Sample SC-R6 contained *Betula* (birch) (0.7 mg), conifer (3.7 mg), unidentified hardwood (<1 mg), and unidentifiable charcoal (2.5 mg). We suggest using the conifer for dating because it was the largest portion of the sample. It may be combined with the birch and unidentified charcoal to boost sample size. Other charred botanical remains included two unidentified charred seeds and some fragments of PET.

Sample SC-R9 contained conifer (24.1 mg), *Pinus* (pine) 3-5 needle type (28.1 mg), and unidentifiable charcoal. We suggest dating the *Pinus* charcoal, although it should be noted that many of the pine species in this group are very long-lived, including *Pinus flexilis*, which lives an average of approximately 1000 years. Ponderosa pine has an average lifespan of 300–600 years, with a maximum of approximately 900 years, while western white pine has an average lifespan of 200-500 years. These two groups of *Pinus* charcoal have similar maximum

longevity values, indicating that any of the pine charcoal may be dated and the two groups of pine charcoal may be combined to obtain a date. This sample also included charred parenchymous tissue consistent with that observed in roots.

Sample SC-R11 contained 2.3 mg conifer charcoal, as well as unidentified hardwood charcoal (0.9 mg), and unidentifiable charcoal (1.2 mg). Of these we suggest using the conifer for dating. If a larger sample size is desired, the conifer, unidentified hardwood, and unidentifiable charcoal may be combined. However, this sample also contained charred *Lycopodium* (club moss) remains (3.8 mg), which could provide a more precise date. Other charred constituents included a Brassicaceae (mustard family) seed fragment, three unidentified seed fragments, and PET.

Unit 3

Unit 3, which mostly comprises silt, is likely contemporaneous with Unit C2. This unit also includes a buried A horizon (Unit 3S). Only one sample from Unit 3 was submitted (SC-R19). Sample SC-R19 contained 7.9 mg of conifer charcoal, likely *Artemisia* (sagebrush) charcoal (1.2 mg), unidentified hardwood charcoal (<1 mg), charred bark (0.4 mg), and unidentifiable charcoal (1.2 mg). We suggest using the conifer charcoal, which has a larger mass, for AMS dating. This sample also included some charred *Lycopodium* (club moss), a *Chenopodium* (goosefoot) seed, and a rodent tooth, the latter of which suggests bioturbation.

Unit C1

Unit C1, which postdates earthquake SC1, is a gravely silt with cobble inclusions in the younger, scarp-derived colluvial wedge that is locally deposited atop Units 1, 2, and C2. Six samples were submitted from this unit, five of which contained enough charcoal for AMS analysis. The charcoal from this unit was mostly conifer, including some charcoal that could be more specifically identified as *Juniperus* (juniper) and *Pinus* (pine). Hardwood charcoal also was present in one sample (SC-R7). Charred nutshell fragments consistent with *Corylus* (hazel) were identified in Sample SC-R3.

Sample SC-R3 included 17 total charcoal fragments, some of which were identified as conifer, while others were not identifiable. We suggest combining all of these to obtain a date since the combined charcoal from this sample weighs only 3.1 mg. The only other charred remains in this sample were two nutshell fragments identified as probable *Corylus*.

Sample SC-R7 contained 5.6 mg *Juniperus* (juniper) charcoal and 16.9 mg charcoal that could only be identified as conifer. Either or both of these are acceptable for dating. Other charcoal from this sample included 2.1 mg of unidentified hardwood and 7.1 mg of unidentifiable charcoal. This sample also contained one unidentified charred seed.

Sample SC-R8 contained 2.5 mg of conifer charcoal and 2 mg of unidentifiable and vitrified charcoal. We suggest using the conifer for dating, but this can be combined with the unidentifiable and vitrified fragments, if necessary. Other charred remains from this sample included PET, resin, and unidentified seed fragments.

Sample SC-R16 did not contain sufficient charcoal to date. Only 1.1 mg of conifer charcoal was identified in this sample. This sample also contained a few fragments of PET and one unidentified charred seed fragment.

Sample SC-R17 contained 5.6 mg of conifer charcoal, suggested for dating, and 6.5 mg of unidentifiable charcoal, which could be used to supplement the conifer charcoal. Other charred remains from this sample included PET and a few small unidentified remains of probable seeds.

Sample SC-R22 contained large amounts of *Juniperus* (juniper) and *Pinus* (pine) charcoal, either of which are adequate for dating. We suggest using the largest single piece of juniper (broken in two for identification), which weighs 12.6 mg. The remaining fragments of juniper weigh a combined 35.7 mg. Based on the geographic and ecological setting, this charcoal is most likely either *Juniperus communis* (common juniper) or *Juniperus scopulorum* (Rocky Mountain juniper). These species live an average of 100-120 years and 200-300 years (2000 years maximum), respectively. This is lower than the average of some of the local pine species, but on par with others. There also were approximately 20 fragments of pine charcoal weighing 21.8 mg and about 40 fragments of conifer charcoal that could not be more specifically identified weighing 41.8 mg. Eighteen fragments of charcoal weighing 9.4 mg were unidentifiable. The only other charred botanical remains were five fragments of parenchymous tissue, which we do not suggest for dating.

Unit Cx

Unit Cx is a silty gravel with cobbles and scarp-derived colluvium in the eastern fault zone. It is deposited atop Unit 1 and possibly corresponds with Unit C1. Only one sample was submitted from this unit (SC-R23). Sample SC-R23 contained 5 mg of conifer charcoal (which we suggest for dating), and 3.4 mg of unidentifiable charcoal. If a larger sample is desired for radiocarbon dating the conifer and unidentifiable charcoal may be combined. Other charred remains from this sample included a likely Brassicaceae (mustard family) seed fragment, an unidentified seed fragment, and one fragment of PET (likely fruit).

SUMMARY AND CONCLUSIONS

Thirty-two bulk sediment samples from the Arentson Gulch and Sheep Creek sites, in the Lost River Range of Custer County, Idaho, were analyzed for macrofloral remains to isolate and identify charred botanical remains suitable for radiocarbon dating. Of these 32 samples, only 19 contained charcoal in amounts sufficient for radiocarbon dating. Twelve of these samples contain quantities of charcoal considered either too small or of marginal size for AMS radiocarbon dating, while seven are of sufficient size for any lab to date. The 12 smaller samples could be processed for AMS radiocarbon dating at PRI, but are likely to be refused by some other labs as too small or fragmentary.

TABLE 1
PROVENIENCE DATA FOR SAMPLES FROM ARENTSON GULCH AND THE
SHEEP CREEK TRENCH, LOST RIVER RANGE, CUSTER COUNTY IDAHO

Sample No.	Unit	Wall	Depth (cmbs)	Provenience/ Description	Analysis
Arentson Gulch					
AG-R6	2	S	107	(AF silt); dense, cemented silt	Bulk sediment
AG-R9		N	157	Back rotated and buried by C2	Bulk sediment
AG-R5	C2S	N	109	Bulk sediment	Bulk sediment
AG-R7		S	113	Bulk sediment	Bulk sediment
AG-R8	C2	S	144	Silt over course, mid-lower wedge	Bulk sediment
AG-R1	3	S	128	Dense silt	Bulk sediment
AG-R10	C1	S	73	Silt-rich colluvial sediment	Bulk sediment
AG-R11			61	Organic rich silt in C1	Bulk sediment
AG-R3	4	S	98	Dense to loose silt	Bulk sediment
AG-R4	5	S	59	Dense organic silt with gravel	Bulk sediment
Sheep Creek Trench					
SC-R1	1	S	323	Gravel in footwall	Bulk sediment
SC-R2			89	Gravel within unit 1Bk	Bulk sediment
SC-R12			123	Below unit, fan soil (1Bk)	Bulk sediment
SC-R13		N	107	Soil in top of unit (below C2)	Bulk sediment
SC-R14		S	154	Top of fan (unit 1) below C1	Bulk sediment
SC-R15			77	Soil in top of unit (below C1)	Bulk sediment
SC-R18			77	Soil on fan (Unit 1) beneath C1	Bulk sediment
SC-R21			329	Lowest in hanging wall gravel	Bulk sediment
SC-R24	2	S, HW	94	Well indurated silt with gravel	Bulk sediment
SC-R4	C2	S	123	Silt pocket, middle of unit	Bulk sediment
SC-R5			131	Silt pocket, base of unit	Bulk sediment
SC-R6			62	Soil on unit	Bulk sediment
SC-R9		N	68	Soil on unit	Bulk sediment
SC-R11			110	Lower middle of unit (between fan & C1)	Bulk sediment
SC-R19	3	S	52	Organic-rich silt (Unit 3) capping C1	Bulk sediment

TABLE 1 (Continued)

Sample No.	Unit	Wall	Depth (cmbs)	Provenience/ Description	Analysis
SC-R7	C1	N	87	Base of unit	Bulk sediment
SC-R3		S	68	Intact block in middle of unit	Bulk sediment
SC-R8			46	Silty base of unit, above C2	Bulk sediment
SC-R16			83	Middle of unit	Bulk sediment
SC-R17			40	Top of unit	Bulk sediment
SC-R22		N, HW	51	Basal unit; brown silt matrix w/ cobbles, gravel	Bulk sediment
SC-R23	Cx	N, FW	52	Brown silt with gravel cobbles	Bulk sediment

TABLE 2
MACROFLORAL REMAINS FROM ARENTSON GULCH,
LOST RIVER RANGE, CUSTER COUNTY IDAHO

Sample Number	Identification	Part	Charred		Uncharred		Weights
			W	F	W	F	Comments
AG-R6	Liters Floated						0.8 L
Unit 2	Light Fraction Weight						2.1673 g
S Wall 107 cmbs	FLORAL REMAINS:						
	Resin	Rootlets		X			Few
	Unidentified					X	Abundant
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Rocks, sand, gravel					X	Abundant
AG-R9	Liters Floated						1.0 L
Unit 2	Light Fraction Weight						0.1478 g
N Wall 157 cmbs	FLORAL REMAINS:						
	Unidentified	Rootlets				X	Few
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Rocks, sand, gravel					X	Abundant
AG-R5	Liters Floated						1.1 L
Unit C2S	Light Fraction Weight						5.4748 g
N Wall 109 cmbs	FLORAL REMAINS:						
	Resin	Rootlets		X		X	Moderate
	Unidentified					X	Abundant
	Unidentified, misc.					X	Few
	Unidentified, vitrified		Seed coat		1		< 0.001 g
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						N/A
	Conifer**	Charcoal		23			0.0028 g
	Unidentifiable, friable**	Charcoal		10*			0.0012 g
	Combine for AMS**						0.0040 g
	NON-FLORAL REMAINS:						
Faunal	Bone				3	0.0024 g	
Rocks, sand, gravel					X	Abundant	

TABLE 2 (Continued)

Sample Number	Identification	Part	Charred		Uncharred		Weights
			W	F	W	F	Comments
AG-R7	Liters Floated						0.8 L
Unit C2S	Light Fraction Weight						1.1806 g
S Wall 113 cmbs	FLORAL REMAINS:						
	Resin	Seed		X			Few
	Unidentified					2	< 0.001 g
	Unidentified					X	Moderate
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer	Charcoal		9			0.0005 g
	NON-FLORAL REMAINS:						
	Faunal	Bone				2	0.0019 g
	Fibers (modern)					X	Few
	Sclerotia					X	Few
	Sand and mineral inclusions					X	Abundant
AG-R8	Liters Floated						0.8 L
Unit C2	Light Fraction Weight						0.6062 g
S Wall 144 cmbs	FLORAL REMAINS:						
	Conifer	Needle	1				0.0002 g
	Unidentified	Seed				1	< 0.001 g
	Unidentified	Rootlets				X	Moderate
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer	Charcoal		1*			< 0.001 g
	NON-FLORAL REMAINS:						
	Faunal	Bone				1	0.0003 g
	Sand and mineral inclusions					X	Abundant

TABLE 2 (Continued)

Sample Number	Identification	Part	Charred		Uncharred		Weights
			W	F	W	F	Comments
AG-R1	Liters Floated						0.95 L
Unit 3 S Wall 128 cmbs	Light Fraction Weight						1.2125 g
	FLORAL REMAINS:						
	Resin	Rootlets Botanics		X		X	Few
	Unidentified					X	Abundant
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer	Charcoal		20*			0.0015 g
	NON-FLORAL REMAINS:						
Rocks, sand, gravel					X	Moderate	
AG-R10	Liters Floated						0.95 L
Unit C1 S Wall 73 cmbs	Light Fraction Weight						2.3252 g
	FLORAL REMAINS:						
	cf. Parenchymous tissue**	Rootlets Botanics		5			0.0020 g
	Resin			X		X	Moderate
	Unidentified					X	Abundant
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer	Charcoal		4			< 0.001 g
Unidentifiable	Charcoal		8*			0.0005 g	
NON-FLORAL REMAINS:							
Rocks, sand, gravel					X	Abundant	

TABLE 2 (Continued)

Sample Number	Identification	Part	Charred		Uncharred		Weights	
			W	F	W	F	Comments	
AG-R11	Liters Floated						0.9 L	
Unit C1 S Wall 61 cmbs	Light Fraction Weight						3.4949 g	
	FLORAL REMAINS:							
	Resin	Rootlets		X		X	Few	
	Unidentified					X	Abundant	
	CHARCOAL/WOOD:							
	Total charcoal ≥ 2 mm						N/A	
	Conifer**	Charcoal		2			< 0.001 g	
	cf. Conifer, vitrified**	Charcoal		6			0.0015 g	
	Unidentifiable**	Charcoal		15*			0.0012 g	
	Combine for AMS**						0.0026 g	
NON-FLORAL REMAINS:								
	Rocks, sand, gravel					X	Moderate	
AG-R3	Liters Floated						0.7 L	
Unit 4 S Wall 95 cmbs	Light Fraction Weight						1.2025 g	
	FLORAL REMAINS:							
	Resin	Seeds		X			Few	
	Unidentified			3			0.0020 g	
	Unidentified		Rootlets				X	Abundant
	Unidentified, misc.	Tissue		4			0.0006 g	
	Unidentified, misc.	Botanics				X	Few	
	CHARCOAL/WOOD:							
	Total charcoal ≥ 2 mm						N/A	
	Conifer**	Charcoal		20*			0.0017 g	
	Unidentified hardwood**	Charcoal		2			0.0006 g	
	Combine for AMS**						0.0023 g	
	NON-FLORAL REMAINS:							
		Sclerotia				X		Few
		Rocks, sand, gravel					X	Abundant

TABLE 2 (Continued)

Sample Number	Identification	Part	Charred		Uncharred		Weights
			W	F	W	F	Comments
AG-R4	Liters Floated						0.9 L
Unit 5	Light Fraction Weight						2.9431 g
S Wall 59 cmbs	FLORAL REMAINS:						
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	<i>Artemisia</i> **	Charcoal		4			0.0031 g
	Conifer	Charcoal		15*			0.0015 g
	Unidentified hardwood	Charcoal		4			0.0005 g
	Unidentifiable, friable	Charcoal		10*			0.0006 g
	NON-FLORAL REMAINS:						
	Sclerotia				X		Few
	Rocks, sand, gravel					X	Few

W = Whole

F = Fragment

X = Presence noted in sample

L = Liter

g = grams

mm = millimeters

* = Estimated frequency

**= Suggested submission for AMS ^{14}C Dating

TABLE 3
INDEX OF MACROFLORAL REMAINS RECOVERED FROM ARENTSON GULCH,
LOST RIVER RANGE, CUSTER COUNTY IDAHO

Scientific Name	Common Name
FLORAL REMAINS:	
Conifer	Cone-bearing, gymnospermous trees and shrubs, mostly evergreens, including the pine, spruce, fir, juniper, cedar, yew, hemlock, redwood, and cypress
Parenchymous tissue	Relatively undifferentiated tissue composed of many similar cells with thin primary walls—occurs in different plant organs in varying amounts, especially large fleshy organs such as roots and stems, but also fruits, seeds, cones, periderm (bark), leaves, needles, etc.
Resin	A solid or highly viscous substance derived from a plant, similar to sap.
Unidentified	Botanical plant remains
CHARCOAL/WOOD:	
<i>Artemisia</i>	Sagebrush
Conifer	Cone-bearing, gymnospermous trees and shrubs, mostly evergreens, including the pine, spruce, fir, juniper, cedar, yew, hemlock, redwood, and cypress
Unidentified	Wood from a tree or shrub
Unidentified hardwood	Wood from a broad-leaved flowering tree or shrub
Unidentified - vitrified	Wood from a tree or shrub, exhibiting a shiny, glassy appearance
NON-FLORAL REMAINS:	
Faunal	Unidentified bone fragments from an animal
Fibers	Modern fibers
Sclerotia	Resting structures of mycorrhizae fungi

TABLE 4
MACROFLORAL REMAINS FROM THE SHEEP CREEK TRENCH,
LOST RIVER RANGE, CUSTER COUNTY IDAHO

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R1	Liters Floated						0.5 L
Unit 1	Light Fraction Weight						0.2608 g
S Wall 323 cmbs	FLORAL REMAINS:						
	Resin	Rootlets		X		X	Few
	Unidentified					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Rock, sand, gravel					X	Abundant
	SC-R2	Liters Floated					
Unit 1	Light Fraction Weight						0.0523 g
S Wall 89 cmbs	FLORAL REMAINS:						
	Unidentified	Rootlets				X	Few
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Unidentifiable < 1 mm	Charcoal		5*			< 0.001 g
	NON-FLORAL REMAINS:						
	Fibers/plastics (modern)	Eggs				X	Few
	Insect					X	Moderate
Rocks, sand, gravel					X	Moderate	

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R13	Liters Floated						0.9 L
Unit 1 N Wall 107 cmbs	Light Fraction Weight						1.6521 g
	FLORAL REMAINS:						
	Resin	Rootlets		X		X	Few
	Unidentified			X		X	Abundant
	Unidentified, misc.					X	Moderate
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Unidentifiable, friable	Charcoal		10*			0.0005 g
	NON-FLORAL REMAINS:						
	Insect	Carapace				X	Few
	Insect	Egg				X	Few
	Plastic, modern					X	Few
Rocks, sand, gravel, mineral inclusions					X	Moderate-Abundant	
SC-R14	Liters Floated						0.8 L
Unit 1 S Wall 154 cmbs	Light Fraction Weight						2.0060 g
	FLORAL REMAINS:						
	Moss	Seed Rootlets Botanics		1		X	Few
	Unidentified						< 0.001 g
	Unidentified					X	Abundant
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Beetle	Carapace/body			X	X	Few
	Faunal	Bone				X	Moderate
	Insect	Carapace			X	X	Few
Insect	Egg			X		Few	
Rocks, sand, gravel					X	Abundant	

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R15	Liters Floated						1.0 L
Unit 1 S Wall 77 cmbs	Light Fraction Weight						2.2715 g
	FLORAL REMAINS:						
	PET			11			0.0008 g
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer**	Charcoal		30*			0.0030 g
	Unidentifiable	Charcoal		15*			0.0017 g
	NON-FLORAL REMAINS:						
	Fibers (modern)					X	Few
	Insect	Carapace				X	Few
	Rocks, sand, gravel					X	Moderate
SC-R18	Liters Floated						0.85 L
Unit 1 S Wall 77 cmbs	Light Fraction Weight						6.5451 g
	FLORAL REMAINS:						
	Parenchymous tissue			9			0.0019 g
	PET			1			< 0.001 g
	Resin			X			Few
	Unidentified	Rootlets				X	Abundant
	Unidentified	Seed/nut		2			0.0009 g
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						0.0029 g
	Conifer	Charcoal		35*			0.0075 g
	Conifer**	Charcoal		5			0.0064 g
	<i>Juniperus</i>	Charcoal		3*			0.0015 g
	Unidentifiable	Charcoal		13			0.0015 g
	NON-FLORAL REMAINS:						
	Rocks, sand, gravel					X	Moderate

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R12	Liters Floated						1.0 L
Unit 1Bk S Wall 123 cmbs	Light Fraction Weight						1.9908 g
	FLORAL REMAINS:						
	Moss	Seeds Rootlets Botanics		2		X	Moderate
	Unidentifiable						< 0.001 g
	Unidentified					X	Moderate
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Faunal	Bone				20*	0.0287 g
Rocks, sand, gravel					X	Moderate	
SC-R21	Liters Floated						0.8 L
Unit 1 S Wall 329 cmbs	Light Fraction Weight						0.0690 g
	FLORAL REMAINS:						
	Unidentified	Rootlets				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Rocks, sand, gravel					X	Abundant
SC-R24	Liters Floated						0.8 L
Unit 2 S, HW Wall 94 cmbs	Light Fraction Weight						1.9191 g
	FLORAL REMAINS:						
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	NON-FLORAL REMAINS:						
	Fibers (modern)					X	Few
Rocks, sand, gravel					X	Abundant	

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R4	Liters Floated						0.7 L
Unit C2	Light Fraction Weight						1.9666 g
S Wall 123 cmbs	FLORAL REMAINS:						
	Moss	Fruit		7		X	Few
	PET			2			0.0015 g
	Resin			5	1		< 0.001 g
	Unidentified	Organics					< 0.001g
	Unidentified	Rootlets				X	Moderate
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer, friable**	Charcoal		35*			0.0050 g
	NON-FLORAL REMAINS:						
	Faunal	Bone				6*	0.0030 g
	Insect	Carapace				X	Few
	Snail	Shell			1		< 0.001 g
Rocks, sand, gravel					X	Moderate	
SC-R5	Liters Floated						0.8 L
Unit C2	Light Fraction Weight						2.2904 g
S Wall 131 cmbs	FLORAL REMAINS:						
	PET	Rootlets		1			< 0.001 g
	Unidentified					X	Abundant
	Unidentified, misc.		Botanics			X	Moderate
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer**	Charcoal		20*			0.0022 g
	Unidentified hardwood	Charcoal		1			< 0.001 g
	Unidentifiable (combine w/ conifer for date)**	Charcoal		15*			0.0007 g
	NON-FLORAL REMAINS:						
	Faunal	Bone/tooth			1	25*	0.0561 g
	Insect	Carapace				X	Few
	Insect	Egg				X	Few
Sclerotia			X	X		Few	
Rocks, sand, gravel					X	Moderate	

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R6	Liters Floated						0.8 L
Unit C2	Light Fraction Weight						2.8559 g
S Wall 62 cmbs	FLORAL REMAINS:						
	PET			2			< 0.001 g
	Unidentified	Seed		2			< 0.001 g
	Unidentified	Rootlets.				X	Abundant
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						N/A
	<i>Betula</i> , friable	Charcoal		15*			0.0007 g
	Conifer, friable**	Charcoal		20*			0.0037 g
	Unidentified hardwood	Charcoal		3			< 0.001 g
	Unidentifiable, friable	Charcoal		20*			0.0025 g
	NON-FLORAL REMAINS:						
	Faunal, possible fish	Vertebra				1	0.0008 g
	Insect	Carapace				X	Moderate
	Sclerotia				X		Few
	Rocks, sand, gravel					X	Abundant
SC-R9	Liters Floated						0.75 L
Unit C2	Light Fraction Weight						3.2748 g
N Wall 68 cmbs	FLORAL REMAINS:						
	Parenchymous tissue	Possible root		2			0.0022 g
	Resin			2 (+)		X	Few
	Monocot	Leaf				X	Few
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						0.0323 g
	<i>Pinus</i> , 3-5 needle type**	Charcoal		35*			0.0281 g
	Conifer	Charcoal		30*			0.0241 g
	Unidentifiable, friable	Charcoal		5			0.0026 g
	Unidentified, 0.5–1 mm	Charcoal		45+			0.0148 g
	NON-FLORAL REMAINS:						
	Fungus	Fruiting body	X	X			Few
	Rocks, sand, gravel					X	Moderate

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R11	Liters Floated						0.8 L
Unit C2	Light Fraction Weight						2.0575 g
N Wall	FLORAL REMAINS:						
110 cmbs	Brassicaceae	Seed		1			< 0.001 g
	<i>Lycopodium</i> **			25*			0.0038 g
	PET			2			< 0.001 g
	Unidentified	Seed		3		1	< 0.001 g
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Moderate
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						N/A
	Conifer**	Charcoal		35*			0.0023 g
	Unidentified hardwood	Charcoal		1			0.0009 g
Unidentifiable	Charcoal		15*			0.0012 g	
NON-FLORAL REMAINS:							
	Rock, sand, gravel					X	Moderate
SC-R19	Liters Floated						0.75 L
Unit 3	Light Fraction Weight						5.1441 g
S Wall	FLORAL REMAINS:						
52 cmbs	<i>Chenopodium</i>	Seed	1				< 0.001 g
	<i>Lycopodium</i> **			8			0.0004 g
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
Total charcoal $\geq$ 2 mm						0.0026 g	
cf. <i>Artemisia</i>	Charcoal		2				0.0012 g
Conifer**	Charcoal		40*				0.0079 g
Unidentified hardwood	Charcoal		3				< 0.001 g
Unidentified	Bark		2				0.0004 g
Unidentifiable	Charcoal		25*				0.0012 g
NON-FLORAL REMAINS:							
	Insect	Carapace				X	Few
	Rodent	Tooth				1	0.0049 g
	Rocks, sand, gravel					X	Moderate

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R7	Liters Floated						0.85 L
Unit C1 N Wall 87 cmbs	Light Fraction Weight						4.2838 g
	FLORAL REMAINS:						
	Unidentified	Seed	1				< 0.001 g
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						0.0111 g
	<i>Juniperus</i> **	Charcoal		15*			0.0056 g
	Conifer**	Charcoal		25*			0.0169 g
	Unidentified hardwood	Charcoal		10*			0.0021 g
	Unidentifiable, friable	Charcoal		12*			0.0071 g
	NON-FLORAL REMAINS:						
	Insect	Carapace				X	Moderate
	Rocks, sand, gravel					X	Abundant
SC-R3	Liters Floated						0.65 L
Unit C1 S Wall 68 cmbs	Light Fraction Weight						3.8815 g
	FLORAL REMAINS:						
	cf. <i>Corylus</i>	Nutshell		2			0.0010 g
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Abundant
	CHARCOAL/WOOD:						
	Total charcoal $\geq$ 2 mm						N/A
	Unidentifiable + conifer (combine for dating)**	Charcoal		17			0.0031 g
	NON-FLORAL REMAINS:						
	Insect	Carapace				X	Few
	Rocks, sand, gravel					X	Moderate

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments	
			W	F	W	F		
SC-R8	Liters Floated						0.65 L	
Unit C1	Light Fraction Weight						2.7482 g	
S Wall 46 cmbs	FLORAL REMAINS:							
	PET	Seed		10			0.0137 g	
	Resin			15(+)			0.0018 g	
	Unidentified			6			2	0.0013 g
	Unidentified						X	Abundant
	Unidentified						X	Abundant
	CHARCOAL/WOOD:							
	Total charcoal ≥ 2 mm						N/A	
	Conifer**	Charcoal		20*			0.0025 g	
	Unidentifiable, friable	Charcoal		30*			0.0010 g	
	Unidentifiable, vitrified	Charcoal		4			0.0010 g	
	NON-FLORAL REMAINS:							
	Insect	Carapace				X	Few	
	Snail	Shell					1	< 0.001 g
	Rocks, sand, gravel						X	Abundant
	SC-R16	Liters Floated						0.9 L
	Unit C1	Light Fraction Weight						4.6700 g
S Wall 83 cmbs	FLORAL REMAINS:							
	PET	Seed		3			<0.001 g	
	Unidentified			1			<0.001 g	
	Unidentified						X	Abundant
	Unidentified, misc.			Botanics				X
	CHARCOAL/WOOD:							
	Total charcoal ≥ 2 mm						N/A	
	Conifer	Charcoal		15*			0.0011 g	
	NON-FLORAL REMAINS:							
	Insect	Carapace				X	Few	
	Rocks, sand, gravel						X	Abundant

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/ Comments
			W	F	W	F	
SC-R17	Liters Floated						0.75 L
Unit C1 S Wall 40 cmbs	Light Fraction Weight						4.8980 g
	FLORAL REMAINS:						
	PET	Rootlets		2			< 0.001 g
	Unidentified					X	Abundant
	Unidentified		2	1	1	1	0.0006 g
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						N/A
	Conifer**	Charcoal		25*			0.0056 g
	Unidentifiable, friable	Charcoal		35*			0.0065 g
	NON-FLORAL REMAINS:						
	Insect	Carapace				X	Few
	Sclerotia				X		Few
	Rocks, sand, gravel					X	Moderate
SC-R22	Liters Floated						0.75 L
Unit C1 N, HW Wall 51 cmbs	Light Fraction Weight						5.0320 g
	FLORAL REMAINS:						
	Parenchymous tissue	Rootlets		5			0.0021 g
	Unidentified					X	Abundant
	Unidentified, misc.					X	Few
	CHARCOAL/WOOD:						
	Total charcoal ≥ 2 mm						0.0772 g
	Conifer	Charcoal		40*			0.0418 g
	<i>Juniperus</i>	Charcoal		30*			0.0357 g
	<i>Juniperus</i> **	Charcoal		2*			0.0126 g
	<i>Pinus</i>	Charcoal		20*			0.0218 g
	Unidentifiable	Charcoal		18			0.0094 g
	NON-FLORAL REMAINS:						
	Insect	Carapace				X	Few
Rocks, sand, gravel					X	Moderate	

TABLE 4 (Continued)

Sample No.	Identification	Part	Charred		Uncharred		Weights/
			W	F	W	F	Comments
SC-R23	Liters Floated						0.8 L
Unit Cx	Light Fraction Weight						4.6816 g
N, FW	FLORAL REMAINS:						
Wall 52 cmbs	cf. Brassicaceae	Seed		1			< 0.001 g
	PET	Fruit		1			0.0010 g
	Unidentifiable (shattered)	Seed coat		1			< 0.001 g
	Unidentified	Rootlets				X	Abundant
	Unidentified, misc.	Botanics				X	Few
CHARCOAL/WOOD:							
Total charcoal ≥ 2 mm						N/A	
Conifer**	Charcoal		30*			0.0050 g	
Unidentifiable, friable/vitrified	Charcoal		15*			0.0034 g	
NON-FLORAL REMAINS:							
Insect	Carapace				X	Few	
Unidentified	Shell				1	< 0.001 g	
Rocks, sand, gravel					X	Moderate	

W = Whole

F = Fragment

X = Presence noted in sample

L = Liter

g = grams

mm = millimeters

* = Estimated frequency

**= Suggested submission for AMS ^{14}C Dating

TABLE 5
INDEX OF MACROFLORAL REMAINS RECOVERED FROM THE SHEEP CREEK TRENCH,
LOST RIVER RANGE, CUSTER COUNTY IDAHO

Scientific Name	Common Name
FLORAL REMAINS:	
Brassicaceae	Mustard family
<i>Chenopodium</i>	Goosefoot, Pigweed
Conifer	Cone-bearing, gymnospermous trees and shrubs, mostly evergreens, including the pine, spruce, fir, juniper, cedar, yew, hemlock, redwood, and cypress
<i>Corylus</i>	Hazelnut
<i>Juniperus</i>	Juniper
<i>Lycopodium</i>	Club moss
Monocot	A member of the Monocotyledonae class of Angiosperms, which include grasses, sedges, lilies, and palms
Moss	Tiny leafy-stemmed flowerless plants (class Bryopsida); any of at least 12,000 species of small, spore-bearing land plants (division Bryophyta) distributed throughout the world except in salt water
Parenchymous tissue	Relatively undifferentiated tissue composed of many similar cells with thin primary walls—occurs in different plant organs in varying amounts, especially large fleshy organs such as roots and stems, but also fruits, seeds, cones, periderm (bark), leaves, needles, etc.
PET	Processed edible tissue
PET fruity tissue	Fruity epitheloid tissues; resemble sugar-laden fruit or berry tissue without the seeds, or succulent plant tissue such as cactus pads
Resin	A solid or highly viscous substance derived from a plant, similar to sap.
Unidentified	Botanical plant remains
CHARCOAL/WOOD:	
<i>Artemisia</i>	Sagebrush
<i>Betula</i>	Birch
<i>Juniperus</i>	Juniper
<i>Pinus</i>	Pine

TABLE 5 (Continued)

Scientific Name	Common Name
Unidentified	Wood from a tree or shrub
Unidentified hardwood	Wood from a broad-leaved flowering tree or shrub
Unidentified - vitrified	Wood from a tree or shrub, exhibiting a shiny, glassy appearance
NON-FLORAL REMAINS:	
Beetle	Body fragments from a member of the order Coleoptera
Faunal	Unidentified bone fragments from an animal
Fibers	Modern fibers
Fungus	Fungus fruiting body
Insect carapace	A dorsal (upper) section of the exoskeleton of an unidentified insect
Insect eggs	Egg from an unidentified insect
Plastic	Modern plastics
Rodent	Tooth fragment from a member of the family Rodentia
Sclerotia	Resting structures of mycorrhizae fungi
Snail shell	Snail shell

REFERENCES CITED

- Carlquist, Sherwin
2001 *Comparative Wood Anatomy: Systematic, Ecological, and Evolutionary Aspects of Dicotyledon Wood*. 2nd ed. Springer Series in Wood Science. Springer, Berlin.
- Hoadley, Bruce
1990 *Identifying Wood: Accurate Results with Simple Tools*. The Taunton Press, Inc., Newtown.
- Martin, Alexander C., and William D. Barkley
1961 *Seed Identification Manual*. University of California, Berkeley.
- Matthews, Meredith H.
1979 Soil Sample Analysis of 5MT2148: Dominguez Ruin, Dolores, Colorado. Appendix B. In *The Dominguez Ruin: A McElmo Phase Pueblo in Southwestern Colorado*, edited by Alan D. Reed. Cultural Resource Series No. 7. Bureau of Land Management, Denver.
- Musil, Albina F.
1963 *Identification of Crop and Weed Seeds*. Agricultural Handbook no. 219. U.S. Department of Agriculture, Washington, D.C.
- Schopmeyer, C. S.
1974 *Seeds of Woody Plants in the United States*. Agricultural Handbook No. 450. United States Department of Agriculture, Washington, D.C.
- Schweingruber, Fritz Hans, Annett Borner, and Ernst-Detlef Schulze
2011 *Atlas of Stem Anatomy in Herbs, Shrubs and Trees* Vol I. Springer-Verlag, Berlin Heidelberg.

2013 *Atlas of Stem Anatomy in Herbs, Shrubs and Trees* Vol. II. Springer-Verlag, Berlin Heidelberg.
- Stenholm, Nancy A.
1994 Paleoethnobotanical Analysis of Archaeological Samples Recovered in the Fort Rock Basin. In *Archaeological Researches in the Northern Great Basin: Fort Rock Archaeology since Cressman*, edited by C. M. Aikens, and D. L. Jenkins, pp. 531-560. University of Oregon Anthropological Papers 50. Department of Anthropology and State Museum of Anthropology, University of Oregon, Eugene, Oregon.

USGS Luminescence Geochronology Lab Report on Feldspar Luminescence Ages for Lost River Fault Zone Samples

Arentson Gulch and Sheep Creek near Mackay, ID, 2021
Principal Investigators: Chris DuRoss

Report prepared by:	Laboratory Director
Harrison J. Gray	Shannon A. Mahan

August 16, 2021

Overview

The following report summarizes the dating results from Paleoseismic trenching within the Lost River Fault Zone near Mackay, ID. Within this report, we detail the methodology used by the USGS Luminescence Geochronology Laboratory to obtain ages including sample preparation methods, luminescence measurement, equivalent dose determination, and dating-related calculations. We recommend that this report be included as the supplementary material for any publication(s) that use the ages within this report. This version supersedes all previous age estimates and reports.

This report contains:

1. A main report synthesizing the results (this document)
2. A .xlsx spreadsheet with the main summary table
3. A .xlsx file with all results in spreadsheet form with a column definitions text file

Disclaimer: Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Contents

1	Executive Summary – H. Gray	3
2	Age Results	3
2.1	SC-L4 Results	5
2.2	SC-L9 Results	7
2.3	AG-L13 Results	9
2.4	AG-L15 Results	11
3	Methods	13
3.1	Luminescence dating background	13
3.2	Sample preparation	13
3.3	Luminescence Machine Analysis	14
3.4	Equivalent Dose Calculation and Age Models	15
3.5	Dose Rate Estimation, Age Calculation, and Water Content Modeling	16
4	References	19
5	Supplemental Material	20
5.1	R code used to calculate ages	20

List of Tables

1	Sample data used to calculate dose rates.	4
2	Sample data used to calculate luminescence ages.	4
3	Miscellaneous data used in age calculation.	4
4	Miscellaneous data used in age calculation.	4
5	Single Aliquot Regeneration protocol	14
6	Single Aliquot Regeneration protocol	14
7	Luminescence Reader Specifications	15
8	Luminescence Reader Specifications	15
9	Aliquot acceptance criteria	16

List of Figures

1	SC-L4 D_E KDE and Radial Plot	5
2	SC-L4 luminescence characteristics	6
3	SC-L9 D_E KDE and Radial Plot	7
4	SC-L9 luminescence characteristics	8
5	AG-L13 D_E KDE and Radial Plot	9
6	AG-L13 luminescence characteristics	10
7	AG-L15 D_E KDE and Radial Plot	11
8	AG-L15 luminescence characteristics	12

1 Executive Summary – H. Gray

To jump straight to the ages, see Table 2 on page 4.

The following report gives preliminary ages for samples taken from two paleoseismic trenches in the Lost River Fault zone. Luminescence dating at this location is very challenging due to unfavorable sample characteristics such as very low signal-to-noise ratios during luminescence measurement and poor dose recovery ¹. We found that the quartz OSL exhibited behavior associated with an unstable fast component, effectively a tendency for the age-dating electron trap to 'leak' electrons over time and cause age underestimation. This behavior consisted of criteria described by Steffen et al. (2009) ². As luminescence ages for paleoseismic studies contribute directly into hazard assessments, it is not worth pursuing quartz OSL here due to the high likelihood of producing inaccurate ages. Although Kenworthy et al. (2014) report quartz OSL ages from nearby alluvial fans, it is our opinion that the quartz sampled in the trenches in this study is unsuitable for dating without an supplemental geochronometric method as used by Kenworthy et al. (2014).

Unfortunately, the alternative option, feldspar post-IR IRSL (pIRIR; Li et al., 2014) also demonstrated challenges. These challenges are low-signal-to-noise ratios and variable dose recovery. However, both were largely mitigated with very large aliquots consisting of hundreds of grains. Examination of the feldspar extracts under microscope reveals very atypical looking samples with large proportions of angular grains of dark color and unknown mineralogy. Rare plagioclase and potassium feldspar grains were present. Thermoluminescence produced during preheating shows strange peaks not typical with pure feldspar extracts. Decay curves from the main pIRIR measurement appeared typical and the large aliquots were able to pass dose recovery within 10 percent error of the given dose. Our interpretation from observation of the luminescence is that this unknown dark mineral was not luminescent during dating measurement, but did show some luminescence during preheating. Next, the improvements in signal-to-noise ratio, despite still being much lower than usual, and dose recovery with increasing aliquot size suggests that the pIRIR luminescence and ages represent data from only a handful of rare grains. Due to these issues, we are only confident that usable luminescence ages will come from the basal and older alluvial fan material. The resolution of ages from younger deposits, include from any colluvial wedges or soils, may have significant measurement resolution problems to argue against their usage. This is also considering the high probability of partial bleaching. Finally, we recommend the use of MAM ages here considering the probability of partial bleaching. We used a σ_b value of 0.08.

2 Age Results

¹Dose recovery is a test we perform to see if a luminescence measurement protocol will return a previously given amount of radiation dose and is a basic test to see if the sample is capable of storing information on age

²low fast to medium OSL component ratios, dose recovery underestimations, and a 'shoulder' on the thermoluminescence produced during preheating

Lab Code	Elevation (m)	Depth (m)	Water Content ^a	K ^b (%)	U ^b (ppm)	Th ^b (ppm)	D_R^c (Gray/ka)
SC-L4	2388	1.2	1 (34) [7]	1.91 ± 0.05	3.74 ± 0.16	9.88 ± 0.47	4.33 ± 0.36
SC-L9	2388	1.3	2 (32) [6]	2.36 ± 0.07	4.27 ± 0.21	9.25 ± 0.36	4.86 ± 0.37
AG-L13	2300	2.1	1 (21) [4]	1.51 ± 0.05	4.06 ± 0.16	5.79 ± 0.47	3.87 ± 0.43
AG-L15	2300	1.4	1 (24) [5]	1.23 ± 0.05	3.33 ± 0.16	4.29 ± 0.44	3.23 ± 0.40

^a Percent water content of field sample used for age calculation, number in parentheses represents the saturated water content, square brackets shows the water content used in age calculation (Method in Table 3).

^b Determined by high resolution Ge gamma spectroscopy or ICP-MS

^c Environmental Dose Rate; calculated using the Dose Rate Age Calculator (Durcan et al., 2015).

Table 1: Sample data used to calculate dose rates.

Lab Code	n ^a	CAM Over- dispersion ^b	CAM D_e^c (Gray)	MAM D_e^d (Gray)	CAM Age ^c (ka)	MAM Age ^d (ka)
SC-L4	19 (20)	12%	60.5 ± 1.8	58.3 ± 2.4	14.0 ± 1.2	13.5 ± 1.2
SC-L9	11 (20)	23%	95.9 ± 6.9	80.4 ± 7.5	19.7 ± 2.1	16.6 ± 2.0
AG-L13	12 (20)	40%	147 ± 18	65.3 ± 7.8	38.0 ± 6.2	16.9 ± 2.8
AG-L15	10 (20)	30%	144 ± 14	108 ± 13	44.6 ± 7.1	33.5 ± 5.9

^a Number of aliquots meeting acceptance criteria, parentheses indicate total number of aliquots measured

^b Defined as the statistical dispersion beyond what would be expected for a perfectly bleached sample.

^c Determined using the function calc_CentralDose from the R-Luminescence package. Uncertainty is 2σ .

^d Determined using the function calc_MinDose from the R-Luminescence package. Uncertainty is 2σ .

^e If ages have a measurable fading rate (Table ??), this value represents the fading-corrected age.

Table 2: Sample data used to calculate luminescence ages.

Lab Code	Latitude (°)	Longitude (°)	Analysis Type	Grain Size (μm)	Water Content Method ^a	D_R Method ^b
SC-L4	44.282727	-113.941353	Feldspar IRSL	90 - 250	FracSat	Gamma
SC-L9	44.282727	-113.941353	Feldspar IRSL	90 - 250	FracSat	Gamma
AG-L13	44.248066	-113.918111	Feldspar IRSL	63 - 250	FracSat	Gamma
AG-L15	44.248066	-113.918111	Feldspar IRSL	63 - 250	FracSat	Gamma

^a Type of assumption used for water content history. F.Sat.Avg = average of field and saturated water contents. Nelson-Rittenour2015 = matric potential method (Nelson and Rittenour, 2015). Frac.Sat = fraction of saturation method.

^b Method used for elemental concentrations of K, U, and Th. Gamma = high Resolution Germanium Gamma Spectrometry. ICP-MS = Inductively-Coupled Plasma Mass Spectrometry.

Table 3: Miscellaneous data used in age calculation.

Lab Code	Given Dose	Hold Time (days)	g-value (% per decade)	Correction Used?	Correction Method
SC-L4	50 Gray	7.83	-1.69 ± 2.51	FALSE	Huntley and Lamothe, 2001
SC-L9	50 Gray	6.67	1.58 ± 4.6	FALSE	Huntley and Lamothe, 2001
AG-L13	50 Gray	10.93	31.0 ± -43.7	FALSE	Huntley and Lamothe, 2001
AG-L15	50 Gray	9.75	3.98 ± -43.9	FALSE	Huntley and Lamothe, 2001

^a notes

Table 4: Miscellaneous data used in age calculation.

2.1 SC-L4 Results

Figure 1: Plots of Equivalent Dose for SC-L4. Red lines shows CAM equivalent dose, blue lines show MAM equivalent dose, solid line shows the age, and dashed lines show uncertainty.

We analyzed SC-L4 using Feldspar IRSL on 90-250 μm size grains. Out of 20 aliquots, 19 produced acceptable luminescence characteristics and passed acceptance criteria as listed in Table 9. Aliquots that passed acceptance criteria are plotted as a kernel density estimate and as a radial plot in Figure 1. Application of the central age model (CAM, Galbraith et al., 1999) produced an equivalent dose of 60.5 ± 1.8 Gray, which leads to a CAM age of 14 ± 1.2 ka. In contrast, application of the minimum age model (MAM, Galbraith et al., 1999) produces an equivalent dose of 58.3 ± 2.4 Gray, which leads to a MAM age of 13.5 ± 1.2 ka.

Figure 2: Plots of luminescence characteristics for SC-L4. A) Compilation of the natural luminescence decay curves for each aliquot. B) Compilation of the growth curves for each aliquot generated by the SAR protocol (Table 9). C) Magnitude of test dose response for each SAR cycle. D) Graph of CAM or MAM age as a function of assumed water content. Vertical black lines show the Field water content, the saturated water content, and our chosen (modeled) water content. E,F,G) Histograms of rejection criteria values. H) Histogram of IRSL Depletion Ratios. I) IRSL observed during depletion ratio test. Note H₂ and I not present for all samples.

2.2 SC-L9 Results

Figure 3: Plots of Equivalent Dose for SC-L9. Red lines shows CAM equivalent dose, blue lines show MAM equivalent dose, solid line shows the age, and dashed lines show uncertainty.

We analyzed SC-L9 using Feldspar IRSL on 90-250 μm size grains. Out of 20 aliquots, 11 produced acceptable luminescence characteristics and passed acceptance criteria as listed in Table 9. Aliquots that passed acceptance criteria are plotted as a kernel density estimate and as a radial plot in Figure 3. Application of the central age model (CAM, Galbraith et al., 1999) produced an equivalent dose of 95.9 ± 6.9 Gray, which leads to a CAM age of 19.7 ± 2.1 ka. In contrast, application of the minimum age model (MAM, Galbraith et al., 1999) produces an equivalent dose of 80.4 ± 7.5 Gray, which leads to a MAM age of 16.6 ± 2.0 ka.

Figure 4: Plots of luminescence characteristics for SC-L9. A) Compilation of the natural luminescence decay curves for each aliquot. B) Compilation of the growth curves for each aliquot generated by the SAR protocol (Table 9). C) Magnitude of test dose response for each SAR cycle. D) Graph of CAM or MAM age as a function of assumed water content. Vertical black lines show the Field water content, the saturated water content, and our chosen (modeled) water content. E,F,G) Histograms of rejection criteria values. H) Histogram of IRSL Depletion Ratios. I) IRSL observed during depletion ratio test. Note H_2 and I not present for all samples.

2.3 AG-L13 Results

Figure 5: Plots of Equivalent Dose for AG-L13. Red lines shows CAM equivalent dose, blue lines show MAM equivalent dose, solid line shows the age, and dashed lines show uncertainty.

We analyzed AG-L13 using Feldspar IRSL on 63-250 μm size grains. Out of 20 aliquots, 12 produced acceptable luminescence characteristics and passed acceptance criteria as listed in Table 9. Aliquots that passed acceptance criteria are plotted as a kernel density estimate and as a radial plot in Figure 5. Application of the central age model (CAM, Galbraith et al., 1999) produced an equivalent dose of 147.0 ± 17.7 Gray, which leads to a CAM age of 38 ± 6.2 ka. In contrast, application of the minimum age model (MAM, Galbraith et al., 1999) produces an equivalent dose of 65.3 ± 7.8 Gray, which leads to a MAM age of 16.9 ± 2.8 ka.

Figure 6: Plots of luminescence characteristics for AG-L13. A) Compilation of the natural luminescence decay curves for each aliquot. B) Compilation of the growth curves for each aliquot generated by the SAR protocol (Table 9). C) Magnitude of test dose response for each SAR cycle. D) Graph of CAM or MAM age as a function of assumed water content. Vertical black lines show the Field water content, the saturated water content, and our chosen (modeled) water content. E,F,G) Histograms of rejection criteria values. H) Histogram of IRSL Depletion Ratios. I) IRSL observed during depletion ratio test. Note H_0 and I not present for all samples.

2.4 AG-L15 Results

Figure 7: Plots of Equivalent Dose for AG-L15. Red lines shows CAM equivalent dose, blue lines show MAM equivalent dose, solid line shows the age, and dashed lines show uncertainty.

We analyzed AG-L15 using Feldspar IRSL on 63-250 μm size grains. Out of 20 aliquots, 10 produced acceptable luminescence characteristics and passed acceptance criteria as listed in Table 9. Aliquots that passed acceptance criteria are plotted as a kernel density estimate and as a radial plot in Figure 7. Application of the central age model (CAM, Galbraith et al., 1999) produced an equivalent dose of 144.0 ± 14.3 Gray, which leads to a CAM age of 44.6 ± 7.1 ka. In contrast, application of the minimum age model (MAM, Galbraith et al., 1999) produces an equivalent dose of 108.0 ± 13.3 Gray, which leads to a MAM age of 33.5 ± 5.9 ka.

Figure 8: Plots of luminescence characteristics for AG-L15. A) Compilation of the natural luminescence decay curves for each aliquot. B) Compilation of the growth curves for each aliquot generated by the SAR protocol (Table 9). C) Magnitude of test dose response for each SAR cycle. D) Graph of CAM or MAM age as a function of assumed water content. Vertical black lines show the Field water content, the saturated water content, and our chosen (modeled) water content. E,F,G) Histograms of rejection criteria values. H) Histogram of IRSL Depletion Ratios. I) IRSL observed during depletion ratio test. Note H_2 and I not present for all samples.

3 Methods

3.1 Luminescence dating background

A luminescence age dates the time since a sand grain or aliquot of grains was last exposed to sunlight. A luminescence age is calculated following the age equation:

$$Age = \frac{D_e}{D_R} \quad (1)$$

where D_e is the *equivalent dose* (units of Joules per kilogram, also written as SI units of *Gray* (Gy) named after physicist Louis Harold Gray and *not* named after geochronologist Harrison James Gray), which is the equivalent amount of radiation needed to produce the observed natural luminescence, and D_R is the natural background radiation dose rate (in units of Gray per kiloannum, Gy/ka)). In luminescence dating, we determine the amount of luminescence as a function of absorbed radiation dose. When we measure the natural luminescence of a sand grain, or aliquots of grains, this provides the equivalent dose D_e . We determine the background dose rate, D_R , either through direct measurement or through measurement of radioactive elements (elemental concentrations: K, U, Th) in a given sample. The following paragraphs describe the methods of determining both D_e and D_R and calculating an age following Equation 1 above.

3.2 Sample preparation

We typically obtain ages following the methods described in Nelson et al. (2015) and Gray et al. (2015), which are repeated here for convenience. To start, we extract both the light-exposed and light-protected sections of a given sample. For tube-based samples, we extracted the outer 2cm from both ends of the tube and reserved this sample for water content determination and elemental concentrations if needed. We then extract the inner, light-protected, contents of the tube for further preparation. For block samples, we spray paint the exterior of the block with black spray-paint to mark light exposed sediment and then dismantle the block and reserve the outer 2cm. We then use this outer radius of the block for water content and elemental concentration measurements. The inner light-protected contents of the block now can be used for further preparation. For film canister collected samples, the entire sample is used for water content prior to further preparation and we reserve a small subsample for elemental concentrations. For all types of sample collection, we will use High Resolution Ge Gamma Spectrometry (Gamma) if the PI submits an additional 300 grams of bulk sample. If this bulk sample is not given, we will use Inductively Coupled Plasma Mass Spectrometry (ICP-MS). Generally we prefer Gamma measurements as these average over a larger volume than ICP-MS and give a more representative values. The resulting values are given in Table 1 in this report. We measure water content percents of the samples by measuring the weights of a subsample when dry, when saturated with water, and the weight when the sample is first opened and calculate using: (wet weight - dry weight)/(dry weight). We calculate the water percent for freshly opened samples (Field Moisture) and again after samples are saturated with water in the laboratory to determine maximum water content (Saturation Moisture). The effects of water content and these measurements are shown in the 'luminescence characteristics' plots throughout this report and reported in Table 1.

Next, we treat the light-protected inner extracts with a series of chemical and mechanical separations. First, we submerge samples in a 5N hydrochloric acid (HCl) solution for the greater of either 4 hours, or until the reaction follows to completion. We then rinse the sample and repeat the process with 30% hydrogen peroxide (H_2O_2). We then apply 100 ml of sodium pyrophosphate ($Na_4P_2O_7$), a deflocculant, to

-
1. Preheat (225 °C) for 60 seconds
 2. Stimulation with blue (470 nm) at 125 °C for 100 seconds (L_n)
 3. Test dose beta irradiation
 4. Preheat (225 °C) for 60 seconds
 5. Stimulation with blue (470 nm) at 125 °C for 100 seconds (T_n)
 6. Repeat steps 1–5 with beta regeneration doses (L_x, T_x)
-

Table 5: Single Aliquot Regeneration protocol for quartz OSL equivalent dose determination (Murray and Wintle, 2000, 2003).

disperse clay particles. This sets the stage for our wet sieving process, where we separate the 90-250 μm grain size range in increments of 20 μm . After separation, we dry and weigh the separations and select the dominant (by mass) grain size for further treatment. In the rare case that no size appears dominant, or if less than 15g of sample is available, we combine the separations. We take this sieved separate and perform a magnetic separation with a Frantz Isodynamic Magnetic Separator (Model L-1) following the methods of Porat (2006). The non-magnetic fraction now contains mostly quartz and feldspar sand. To separate quartz from feldspar, we perform a density separation using lithium sodium polytungstate, a liquid with adjustable density such that quartz can sink and feldspars will float. We then etch the isolated quartz fraction with a 49% hydrofluoric (HF) solution to remove feldspar inclusions and any accessory minerals followed by an 5N HCl rinse to remove fluoride precipitates. Finally, aliquots of quartz or feldspar are plated onto stainless steel discs using commercial silicone spray. The discs of quartz and/or feldspar are now ready for machine analysis.

3.3 Luminescence Machine Analysis

We measure the quartz OSL or feldspar IRSL using the single aliquot regeneration (SAR) protocol (Murray and Wintle, 2000; 2003) given in Table 5. See Tables 8 and 7 for machine specifications. Decay curves showing the luminescence output during measurement and growth curves showing the relation between luminescence and equivalent dose are presented in the supplementary materials with this report. These graphs are examined by lab staff during measurement to check for signs of contamination by non quartz or feldspathic minerals or for evidence of luminescence saturation. We attempt to run between 30 to 48 aliquots per sample. However, factors such as sample availability can limit the number of runs performed or if laboratory staff determine that further aliquots are not necessary to calculate an age. The

-
1. Preheat (250-320 °C) for 60 seconds
 2. Stimulation with IR LEDs (870 nm) at 50 °C for 200 seconds
 3. Stimulation with IR LEDs (870 nm) at 225–290 °C for 200 seconds (L_n)
 4. Preheat (250-320 °C) for 60 seconds
 5. Stimulation with IR LEDs (870 nm) at 50 °C for 200 seconds
 6. Stimulation with IR LEDs (870 nm) at 225–290 °C for 200 seconds (T_n)
 7. Hot Bleach with IR LEDs (870 nm) at 325 °C for 40 seconds
 8. Repeat steps 1–5 with beta regeneration doses (L_x, T_x)
-

Table 6: Single Aliquot Regeneration protocol for feldspar post-IR IRSL equivalent dose determination (Li et al., 2014).

Machine:	Automated Risø TL/luminescence-DA-15 and/or DA-20
Irradiation Source (β)	Sr ⁹⁰ /Y ⁹⁰ (0.845 Gy/s)
Stimulation source:	Blue diodes (quartz): emission centered on 470 nm IR diodes (feldspar): emission centered on 870 nm
Power delivered to aliquot:	136 mW/cm ² for quartz OSL (85% power) 144 mW/cm ² for feldspar IRSL (90% power)
Photomultiplier:	Thorn-EMI 9235Q
Detection filters:	two Hoya U340 filters

Table 7: Machine parameters used in preparation and analyses of samples for quartz OSL, feldspar IRSL, and/or feldspar pIRIR for the Freiburg luminescence readers.

Machine:	Freiburg Instruments Lexsyg Research and/or Lexsyg Smart
Irradiation Source (β)	Sr ⁹⁰ /Y ⁹⁰ (0.11 Gy/s)
Stimulation source:	Blue diodes (quartz): LED ENGIN LZ1-00B200 – 458 nm emission IR diodes (feldspar): LED ENGIN LZ1-00R400 – 850 nm emission
Power delivered to aliquot:	Variable 50–100 mW/cm ² for quartz OSL Variable: 100–130 mW/cm ² for feldspar IRSL/pIRSL
Photomultiplier:	Hamamatsu H7360-02
Detection filters:	Variable with automatic filter changer

Table 8: Machine parameters used in preparation and analyses of samples for quartz OSL, feldspar IRSL, and/or feldspar pIRIR for the Risø luminescence readers.

number of aliquots that pass acceptance criteria (Table 9) depends on the mineralogy of the sample and the source geology. Often, samples that contain quartz and/or feldspar derived from volcanic or hydrothermal deposits will have unfavorable luminescence characteristics. The percentage of accepted aliquots varies greatly, but a lower percentage does not imply an inaccurate age. Rather, it indicates a longer sample measuring time as more aliquots are needed to improve age estimates and lower uncertainty. The number of aliquots measured is usually decided on a sample-by-sample basis by laboratory staff in consultation with the director.

3.4 Equivalent Dose Calculation and Age Models

After collecting data of the luminescence of plated aliquots, we analyze the results using the r-Luminescence package (Kreutzer et al., 2012; 2018) for the programming language R (R Core Team, 2018). The exact code used to produce this report is given in the Supplementary Materials at the end of this report. The general workflow is to import the machine files into R, process them using handling functions and analysis functions such as `analyze_SAR.CWOSL()` to perform single aliquot regeneration and then driving the output into analysis functions such as `calc_CentralDose()` and `calc_MinDose()` to obtain the central age model and minimum age model results respectively. Various other parameters are estimated and can be seen in detail in the code below. The final output is the summary tables (Table 1 and Table 2) and the KDE/radial plots throughout this document.

The central age model (CAM) and minimum age model (MAM) used in this report are statistical tools specialized for luminescence dating (Galbraith et al., 1999; Galbraith and Roberts, 2012). They effectively

solve for the dose that all grains received during burial. The central age model is analogous to a weighted average, but is different in that it accounts for the uncertainties associated with the doses received by grains during burial. When a sample has been fully bleached prior to deposition, the CAM will isolate the true burial dose. However, fully bleached samples can be rare in some depositional environments. In such cases where incomplete bleaching is expected, we use the minimum age model (MAM). The MAM effectively weights the lower dose grains based on the idea that lower dosed grains are more likely to have been fully bleached prior to deposition. For fluvial and/or alluvial environments, the MAM is usually preferred. Note that in some cases, the CAM and MAM will produce overlapping results. This occurs when the statistical scatter of the sample is similar to the scatter expected for a well-bleached sample (20%, Cunningham and Wallinga, 2012). Both the CAM and MAM are identifying the central population as the population most likely to represent the true age. If there is overlapping results, we generally prefer the CAM. If they do not overlap, we generally prefer the MAM. However, the decision of which age model to prefer involves multiple factors and varies on a case-by-case basis.

3.5 Dose Rate Estimation, Age Calculation, and Water Content Modeling

To estimate the background dose rate (Table 1) we require measurements of the elemental concentrations of the weight percent of potassium (K%), and the parts per million (ppm) of Uranium (U) and Thorium (Th) as previously discussed. If only tube-based samples are provided, we will extract sample and submit for ICP-MS to determine K, U, and Th values. If we are provided with a large bulk sample of approximately 600 grams, we will use High Resolution Ge Gamma Spectrometry, which we find gives results with comparable or better precision/accuracy and has a faster turn-around time.

We calculate dose rates (1) using the Dose Rate and Age Calculator (DRAC) by Durcan et al. (2015). We use DRAC as it is an open-source calculator that is continuously updated by the authors as improvements in luminescence dating develop. DRAC has been vetted by the community and also handles uncertainty propagation. Its use greatly minimizes the potential for calculation errors. DRAC takes inputs such as elemental concentrations of K, U, and Th, and other sample data and returns an estimate of the environmental dose rate. Our exact input and use of DRAC functions is demonstrated in the R code in the supplemental material. Finally, we calculate ages by using Equation 1 and propagating uncertainty using standard error propagation formulas (Table 2).

One of the largest sources of uncertainty in the dose rate calculation is an assumption of the water-content history of the sample over its burial lifetime. The water content, here defined as the water held in pore spaces within sediment, serves to lower the dose rate by absorbing radiation. This lowered dose rate leads to a decrease in luminescence generated per unit time. However, the water content of a sedimentary unit can change throughout time due to changes in water table height as a function of various processes such as

Recycling Ratio:	10%
Recuperation:	5%
Maximum Test Dose Error:	20%
Maximum Paleodose Error:	20%
IR Depletion Ratio:	10%

Table 9: Acceptance criteria used to filter aliquots for equivalent dose determination. Aliquots passing these criteria are plotted in KDE and radial plots.

climate change. There are three methods available to estimate the lifetime average water content of a sample. First, the *in-situ* water content of a sample can be used to estimate the lifetime water content. Another alternative used in luminescence dating is to choose a fraction of saturated water content, e.g. 10%, 25%, etc. A third alternative we describe below is to model the lifetime water content following Nelson and Rittenour (2015).

We follow Nelson and Rittenour (2015) by constructing the model of the water content by using the van Genuchten (1980) water retention curve for each sample:

$$\theta(h) = \theta_r + \frac{\theta_s - \theta_r}{[1 + (\alpha|h|)^n]^{1-\frac{1}{n}}} \quad (2)$$

where $\theta(h)$ is the modeled volumetric water content (cm^3/cm^3), θ_r is the residual volumetric water content (cm^3/cm^3), θ_s is the saturated water content (cm^3/cm^3), α ($1/\text{cm}^3$) and n (dimensionless) are curve shape parameters, and h is the matric potential of the sediment (cm). We measure θ_s during sample preparation, and using soil texture class average values for θ_r , α , and n from the US Salinity Lab (<https://www.ars.usda.gov/pacific-west-area/riverside-ca/us-salinity-laboratory/docs/rosetta-class-average-hydraulic-parameters/>). We determine the soil texture class (sand, clay, loam, etc) using particle size measurements, laboratory evaluation, or client provided data. To estimate h , we follow Nelson and Rittenour (2015) and estimate the Mean Annual Water State (MAWS) matric potential using the USDA-NRCS Soil Moisture Regime Classification system. The USDA-NRCS Soil Moisture Regime Classification system provides estimates for the matric potential of a local soil, which can then be used with Equation 2 to provide a modeled water content value. See Nelson and Rittenour (2015) for detailed methodology.

The choice of water content method (*in-situ*, fraction of saturation, or modeling) depends on the applicability of the method for each sample. The *in-situ* method assumes that the present water-content value is reflective of the average water content for the lifetime of a sample. In cases where the water content is not expected to have changed significantly following deposition, the *in-situ* method is appropriate. Examples include samples obtained by coring more than a few meters below the surface or for samples below the water table. However, the *in-situ* can often be non-representative of the average water content over the life of sample prior to sampling (Nelson and Rittenour, 2015). For example, if one is sampling an escarpment, the recent exposure of the surface will lead to drying and a lowering of the water table. An *in-situ* sample of water content in this environment may not be representative of the water conditions prior to erosion of the escarpment.

In contrast, the fraction of saturation method avoids the issue of the *in-situ* value being non-representative of the lifetime average value. The fraction of saturation method may be more appropriate when the water content history of a sample is known or expected to have changed over time. An example may be paleo-lake deposits where the sample remained saturated following deposition until the lake dried up and the sample became desiccated for the rest of its history. However, the fraction of saturation method requires the user to assume some value and it may be difficult to robustly choose a value. Individual cases may require further analysis to estimate an appropriate fraction.

Finally, the modeling method, as described above, presents an alternative that seeks to minimize subjectivity in the water content estimation. The modeling approach is appropriate for free-draining samples within 3 meters of the land surface and are not affected by perched or fluctuating water tables (Nelson and Rittenour, 2015). The modeling method assumes that group averages of hydraulic parameters (θ_r , α , and n) are reflective of the hydraulic parameters of the sample. As such, the uncertainty on the

modeled water content value is often large (30-50%). The use of class average hydraulic parameters is used heavily in soil science to general success (Vereecken et al., 2010). In addition, the method assumes that the Mean Annual Water State (MAWS) matric potential using the USDA-NRCS Soil Moisture Regime Classification system is approximately appropriate for the sample locations. The USDA-NRCS system is fairly coarse in resolution and may not capture local variations in sample matric potential. As with the *in-situ* and fraction of saturation methods, the modeling approach requires a consideration of its base assumptions and applicability on a sample-by-sample basis.

4 References

- Bell W (1979). Attenuation factors for the absorbed radiation dose in quartz inclusions for thermoluminescence dating. *Ancient TL* 8, 1-12.
- Brennan B, Lyons R, Phillips S (1991). Attenuation of alpha particle track dose for spherical grains. *International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements*, 18, 249-253.
- Cunningham, A. C., & Wallinga, J. (2012). Realizing the potential of fluvial archives using robust OSL chronologies. *Quaternary Geochronology*, 12, 98-106.
- Cunningham, A. C., Wallinga, J., & Minderhoud, P. S. (2011). Expectations of scatter in equivalent-dose distributions when using multi-grain aliquots for OSL dating. *Geochronometria*, 38(4), 424.
- Durcan, J.A., King, G.E., and Duller, G.A.T., 2015. DRAC: Dose rate and age calculator for trapped charge dating. *Quaternary Geochronology*, 28, 54-61.
- Galbraith, R. F., Roberts, R. G., Laslett, G. M., Yoshida, H., & Olley, J. M. (1999). Optical dating of single and multiple grains of quartz from Jinmium rock shelter, northern Australia: Part I, experimental design and statistical models. *Archaeometry*, 41(2), 339-364.
- Galbraith, R. F., & Roberts, R. G. (2012). Statistical aspects of equivalent dose and error calculation and display in OSL dating: an overview and some recommendations. *Quaternary Geochronology*, 11, 1-27.
- Gray, H. J., Mahan, S., Rittenour, T. M., & Nelson, M. S. (2015). Guide to luminescence dating techniques and their application for paleoseismic research. In *Basin and Range Province Seismic Hazards Summit III*, 2015.
- Kenworthy, M. K., Rittenour, T. M., Pierce, J. L., Sutfin, N. A., & Sharp, W. D. (2014). Luminescence dating without sand lenses: An application of OSL to coarse-grained alluvial fan deposits of the Lost River Range, Idaho, USA. *Quaternary Geochronology*, 23, 9-25.
- Kreutzer, S., Dietze, M., Burow, C., Fuchs, M. C., Schmidt, C., Fischer, M., & Friedrich, J. (2017). Luminescence: Comprehensive luminescence dating data analysis. R package version 0.8.5.
- Kreutzer S, Schmidt C, Fuchs MC, Dietze M, Fischer M, Fuchs M (2012). Introducing an R package for luminescence dating analysis. *Ancient TL*, 30(1), 1-8.
- Murray, A. S., & Wintle, A. G. (2000). Luminescence dating of quartz using an improved single-aliquot regenerative-dose protocol. *Radiation measurements*, 32(1), 57-73.
- Murray, A. S., & Wintle, A. G. (2003). The single aliquot regenerative dose protocol: potential for improvements in reliability. *Radiation measurements*, 37(4-5), 377-381.
- Nelson, M. S., Gray, H. J., Johnson, J. A., Rittenour, T. M., Feathers, J. K., & Mahan, S. A. (2015). User guide for luminescence sampling in archaeological and geological contexts. *Advances in Archaeological Practice*, 3(2), 166-177.
- Nelson, M. S., & Rittenour, T. M. (2015). Using grain-size characteristics to model soil water content: Application to dose-rate calculation for luminescence dating. *Radiation Measurements*, 81, 142-149.
- Porat, N. (2006). Use of magnetic separation for purifying quartz for luminescence dating. *Ancient TL*, 24(2), 33-36.
- Prescott J, Hutton J (1994). Cosmic ray contributions to dose rates for luminescence and ESR dating: Large depths and long-term time variations. *Radiation Measurements*, 23, 497-500.
- Steffen, D., Preusser, F., & Schlunegger, F. (2009). OSL quartz age underestimation due to unstable signal components. *Quaternary Geochronology*, 4(5), 353-362.
- Van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils 1. *Soil science society of America journal*, 44(5), 892-898.
- Vereecken, H., Weynants, M., Javaux, M., Pachepsky, Y., Schaap, M. G., & Genuchten, M. T. (2010). Using pedotransfer functions to estimate the van Genuchten-Mualem soil hydraulic properties: A review. *Vadose Zone Journal*, 9(4), 795-820.

5 Supplemental Material

5.1 R code used to calculate ages

```
1 ## USGS R script for Luminescence analysis
2 # Written by H GRAY 2018-2019
3
4 ## TO DO:
5
6 # create dataframe space for unfaded ages using "-" if they are not corrected
7
8 # create LaTeX table of faded and unfaded ages:
9 #   Name | Dose Given | Hold Time | Gvalue | Uncorrected CAM age | Corrected CAM Age | Uncorrected MAM age | Corrected MAM Age
10
11 # rearrange the luminescence characteristics figure
12
13 # draw lines on growth curves in lum characteristics figure
14
15 # bibtex citations
16
17 # explain rejection criteria in TeX report doc
18
19 #####
20
21 # Luminescence tests: preheat plateau/dose recovery, fading, Feldspar unbleachable dose,
22 # during measurement: recuperation, recycling ratio, error limits, 2D0 test
23
24 # recycling ratio tests that the sensitivity correction in the SAR cycle is working
25 # recuperation tests for the photo- or thermal-transferrence of charge during an OSL readout. can sometimes be corrected with a
26 # thermal wash at the end of a SAR cycle
27
28 # 2D0 test ensures that the aliquot is not skewed by the presence of saturated grains
29 #####
30
31 # to see the source code for any function, use getAnywhere()
32
33 # print(data.frame(names = list.files(pattern = "*.bin")),row.names = FALSE,right=F)
34
35 #suppressPackageStartupMessages
36
37 #
38 # JOTR_1_5_ages = c(34.703,19.501,4.609,21.264,19.309)
39 # JOTR_1_5_ages_err = c(3.721,2.183,0.509,2.344,1.876)
40 #
41 for (i in 1:5){
42 #   calc_FadingCorr(age.faded = c(JOTR_1_5_ages[i],JOTR_1_5_ages_err[i]),
43 #                   g_value = c(4.5, 0.5),
44 #                   tc = 820,
45 #                   txtProgressBar = TRUE)
46 # }
47 #
48 #
49 #####
50
51 ## Initialize
52
53 rm(list = ls())
54
55 library("Luminescence")
56 library("openxlsx")
57 library("xtable")
58 library("stringr")
59 library("filestrings")
60 library("ggplot2")
61 library("ggplotify")
62 library("gridExtra")
63
64 # source("C:\\Users\\hgray\\OneDrive - DOI\\Gray_PortaOSL-ColluvialWedges\\Lum_dating_analysis\\Original BIN files\\Quartz\\bin_
65 #   file_management.R")
66
67 # calc_gSGC()
68 # plot_KDE()
69
70 #####
71
72 ## Core inputs
73
74 wdir = "C:\\Users\\hgray\\DOI\\USGS Luminescence Lab - General\\Lab Projects\\DuRoss, Briggs, Gold-Arnetson Gulch and Sheep Creek
75 \\Lum_dating_analysis\\"
76 setwd(wdir)
77
78 # print(data.frame(names = list.files(pattern = "*.bin")),row.names = FALSE)
79
80 input_data = read.xlsx("input_table.xlsx", colNames = TRUE, rowNames = FALSE, skipEmptyRows = TRUE)
81 #input_data = input_data[complete.cases(input_data),]
```

```

82 bootstrap_multi_core_mam_true_or_false = FALSE #TRUE
83
84 #####
85
86 ## Basic starting routines, build summary table
87
88 num_samples = nrow(input_data)
89 sample_list = gsub(pattern=" ",x = input_data$Name, replacement = "-")
90 depth_of_samples = input_data$depth # sample depths in meters
91 field_water_content_of_samples = input_data$water_field
92 saturation_water_content_of_samples = input_data$water_sat
93
94 soil_texture = input_data$soil_texture
95 soil_moisture_regime = input_data$soil_moisture_regime
96
97 mesh_size_min = input_data$mesh_min
98 mesh_size_max = input_data$mesh_max
99
100 Th_values = round(input_data$Th,3)
101 Th_err = round(input_data$Th_err,3)
102
103 U_values = round(input_data$U,3)
104 U_err = round(input_data$U_err,3)
105
106 K_values = round(input_data$K,3)
107 K_err = round(input_data$K_err,3)
108
109 latitude_values = input_data$lat
110 longitude_values = input_data$long
111 elevation_values = round(input_data$elevation)
112
113 machine = input_data$Machine
114 OSL_IRSL_type = input_data$analysis_type
115 mineral = input_data$mineral
116 analysis_type = paste(mineral, OSL_IRSL_type, sep=" ")
117 bin_or_xlsx = input_data$bin_or_xlsx
118 file_names = input_data$file_name_plus_extension
119
120 DR_type = input_data$DR_type
121
122 sigma_b_input = input_data$sigma_b
123 MAM_log = input_data$MAM_log
124
125 water_type = input_data$water_type
126 frac_sat_value = input_data$water_frac_sat
127 frac_sat_value_err = input_data$water_frac_sat_err
128
129 sig_min = input_data$signal_min
130 sig_max = input_data$signal_max
131 background_min = input_data$background_min
132 background_max = input_data$background_max
133 fit_eqn = input_data$fit_method
134 Recyc_Ratio = input_data$RecycleRatio
135 Recup_values = input_data$Recup
136 PaleoErr = input_data$PaleoErr
137 TestErr = input_data$TestErr
138 ExceedMax = input_data$ExceedMax
139 SAR_num_vals = input_data$SAR_num
140 use_FMM = input_data$UseFMM
141
142 g_vals = input_data$g_value
143 g_value_err = input_data$g_value_err
144 tc_val = input_data$tc_vals
145 hold_time = input_data$hold_time
146 fade_correction_used = input_data$FadingCorrUsed
147 fade_correction_type = input_data$fade_correction_type
148
149 notes = input_data$notes
150
151 ## build grain size arrays and mineral arrays
152
153 grain_size_min = rep(NA,num_samples)
154 grain_size_max = rep(NA,num_samples)
155
156 ## min grain size
157
158 for (grain_iter in c(1:num_samples)){
159
160   if (mesh_size_min[grain_iter]==80){
161     grain_size_min[grain_iter] <- 180
162   }
163   if (mesh_size_min[grain_iter]==100){
164     grain_size_min[grain_iter] <- 150
165   }
166
167   if (mesh_size_min[grain_iter]==120){
168     grain_size_min[grain_iter] <- 125
169   }
170
171   if (mesh_size_min[grain_iter]==170){
172     grain_size_min[grain_iter] <- 90
173   }

```

```

174
175   if (mesh_size_min[grain_iter]==230){
176     grain_size_min[grain_iter] <- 63
177   }
178   if (mesh_size_min[grain_iter]=="combined"){
179     grain_size_min[grain_iter] <- 90
180   }
181
182   if (mesh_size_min[grain_iter]=="silt"){
183     grain_size_min[grain_iter] <- 4
184   }
185 }
186 }
187
188 ## max grain size
189
190 for (grain_iter in c(1:num_samples)){
191
192   if (mesh_size_max[grain_iter]==80){
193     grain_size_max[grain_iter] <- 250
194   }
195   if (mesh_size_max[grain_iter]==100){
196     grain_size_max[grain_iter] <- 180
197   }
198
199   if (mesh_size_max[grain_iter]==120){
200     grain_size_max[grain_iter] <- 150
201   }
202
203   if (mesh_size_max[grain_iter]==170){
204     grain_size_max[grain_iter] <- 125
205   }
206
207   if (mesh_size_max[grain_iter]==230){
208     grain_size_max[grain_iter] <- 90
209   }
210   if (mesh_size_max[grain_iter]=="combined"){
211     grain_size_max[grain_iter] <- 250
212   }
213   if (mesh_size_max[grain_iter]=="silt"){
214     grain_size_max[grain_iter] <- 11
215   }
216 }
217
218 mineral_for_DRAC = rep(NA,num_samples)
219 conversion_factors = rep("X",num_samples)
220 calc_Rb_external = rep("Y",num_samples) # change to account for ICP-MS data in the future
221 calc_Rb_internal = rep("N",num_samples) # change to account for ICP-MS data in the future
222 alpha_attenuation = rep("NA",num_samples)
223 beta_attenuation = rep(NA,num_samples)
224 etch_depth_min = rep(NA,num_samples)
225 etch_depth_max = rep(NA,num_samples)
226 beta_etch_attenuation_factor = rep("Bell1979", num_samples)
227 alpha_track = rep(NA,num_samples)
228 alpha_track_err = rep(NA,num_samples)
229
230 internal_K_percent = rep(0,num_samples)
231
232 for (mineral_iter in c(1:num_samples)){
233   if (mineral[mineral_iter]=="Quartz"){
234     mineral_for_DRAC[mineral_iter] = "Q"
235     alpha_attenuation[mineral_iter] = "Brennanetal1991"
236     beta_attenuation[mineral_iter] = "Guerinetal2012-Q"
237     etch_depth_min[mineral_iter] = 8
238     etch_depth_max[mineral_iter] = 10
239     conversion_factors[mineral_iter] = "Guerinetal2011"
240     alpha_track[mineral_iter] = 0
241     alpha_track_err[mineral_iter] = 0
242   }
243
244   if (mineral[mineral_iter]=="Feldspar"){
245     mineral_for_DRAC[mineral_iter] = "F"
246     alpha_attenuation[mineral_iter] = "Bell1980"
247     beta_attenuation[mineral_iter] = "Guerinetal2012-F"
248     etch_depth_min[mineral_iter] = 0
249     etch_depth_max[mineral_iter] = 0
250     internal_K_percent[mineral_iter] = 12.5
251
252     calc_Rb_internal[mineral_iter] = "Y"
253     conversion_factors[mineral_iter] = "AdamiecAitken1998"
254     alpha_track[mineral_iter] = 0.15
255     alpha_track_err[mineral_iter] = 0.05
256   }
257 }
258
259 if (mineral[mineral_iter]=="Polymneral"){
260   mineral_for_DRAC[mineral_iter] = "PM"
261   alpha_attenuation[mineral_iter] = "Bell1980"
262   beta_attenuation[mineral_iter] = "Mejdahl1979"
263   etch_depth_min[mineral_iter] = 0
264   etch_depth_max[mineral_iter] = 0
265   conversion_factors[mineral_iter] = "AdamiecAitken1998"

```

```

266     alpha_track[mineral_iter] = 0.086
267     alpha_track_err[mineral_iter] = 0.0038
268   }
269 }
270 }
271
272 ## Scaling for gamma dose
273
274 scale_gamma = rep(NA,num_samples)
275
276 for (gamma_iter in c(1:num_samples)){
277   if (depth_of_samples[gamma_iter]<=0.3){
278     scale_gamma[gamma_iter] = "Y"
279   } else {
280     scale_gamma[gamma_iter] = "N"
281   }
282 }
283
284
285
286 num_accepted = rep(NA,num_samples)
287 num_total = rep(NA,num_samples)
288
289 #####
290
291 ## Estimation of the water content based on Nelson and Rittenour 2015
292
293 # Import Rosetta data
294 rosetta_soil_textures = read.xlsx("soil_textures_class_averages.xlsx", colNames = TRUE, rowNames = FALSE, skipEmptyRows = TRUE)
295 rosetta_soil_textures = rosetta_soil_textures[complete.cases(rosetta_soil_textures),]
296 MAWS_array = read.xlsx("MAWS.xlsx", colNames = TRUE, rowNames = FALSE, skipEmptyRows = TRUE)
297
298 #creat dummy arrays
299 sample_soil_params_array = data.frame()
300 sample_MAWS_vals = data.frame()
301
302 # fill arrays with Rosetta values depending on the sample
303 for (water_iter in c(1:num_samples)){
304
305   sample_soil_params_array = rbind(sample_soil_params_array,subset(rosetta_soil_textures, Texture_Class==soil_texture[water_iter]))
306   sample_MAWS_vals = rbind(sample_MAWS_vals, subset(MAWS_array, soil_class==soil_moisture_regime[water_iter]))
307 }
308
309 # add sample names
310 sample_soil_params_array = cbind(sample_list,sample_soil_params_array)
311 sample_MAWS_vals = cbind(sample_list, sample_MAWS_vals)
312
313 # Break up the Van Genuchten Calc into pieces to limit mistakes
314 A = (input_data$water_sat*(1.8/1)*(1/100) - sample_soil_params_array$theta_r)
315 B = (10^sample_soil_params_array$log_alpha) * (abs(sample_MAWS_vals$geo_mean))
316 C = 1 + B^(10^sample_soil_params_array$log_n)
317 D = 1 - 1/(10^sample_soil_params_array$log_n)
318
319 van_Genuchten_WRC = sample_soil_params_array$theta_r + A/(C^D))
320
321
322 # add to variables later added to DRAC (Choose between the Nelson and Rittenour method or something else)
323
324 modeled_water_content_of_samples = round(100*as.numeric(round(van_Genuchten_WRC*(1/1.8),4)))
325 modeled_water_err = 0.3*modeled_water_content_of_samples
326
327 preferred_water_content_of_samples = rep(NA,num_samples)
328 preferred_water_content_of_samples_err = rep(NA,num_samples)
329
330 for (water_iter2 in seq(1,num_samples,1)){
331   if (water_type[water_iter2]=="NelsonRittenour2015"){
332     preferred_water_content_of_samples[water_iter2] = modeled_water_content_of_samples[water_iter2]
333     preferred_water_content_of_samples_err[water_iter2] = modeled_water_err[water_iter2]
334   }
335
336   if (water_type[water_iter2]=="FieldSatAvg"){
337     preferred_water_content_of_samples[water_iter2] = (saturation_water_content_of_samples[water_iter2] + field_water_content_of_
338       samples[water_iter2])/2
339     preferred_water_content_of_samples_err[water_iter2] = (saturation_water_content_of_samples[water_iter2]*0.05 + field_water_
340       content_of_samples[water_iter2]*0.05)/2
341   }
342
343   if (water_type[water_iter2]=="Field"){
344     preferred_water_content_of_samples[water_iter2] = field_water_content_of_samples[water_iter2]
345     preferred_water_content_of_samples_err[water_iter2] = 0.05*field_water_content_of_samples[water_iter2]
346   }
347
348   if (water_type[water_iter2]=="Saturated"){
349     preferred_water_content_of_samples[water_iter2] = saturation_water_content_of_samples[water_iter2]
350     preferred_water_content_of_samples_err[water_iter2] = 0.05*saturation_water_content_of_samples[water_iter2]
351   }
352
353   if (water_type[water_iter2]=="FracSat"){
354     preferred_water_content_of_samples[water_iter2] = frac_sat_value[water_iter2]
355     preferred_water_content_of_samples_err[water_iter2] = frac_sat_value_err[water_iter2]
356   }
357 }

```



```

356 }
357
358 # modeled_water_content_of_samples = (saturation_water_content_of_samples + field_water_content_of_samples)/2
359 # modeled_water_err = (saturation_water_content_of_samples*0.05 + field_water_content_of_samples*0.05)/2
360
361 ## enter in preferred water content
362
363 # preferred_water_content_of_samples = 0.5*saturation_water_content_of_samples #field_water_content_of_samples #
364 # preferred_water_content_of_samples_err = saturation_water_content_of_samples*0.05 #field_water_content_of_samples*0.3
365
366 #####
367
368 # prepare and submit DRAC inputs
369
370 dose_rates <- rep(NA,num_samples)
371 dose_rates_err <- rep(NA,num_samples)
372
373 cosmic_dose_rates <- rep(NA,num_samples)
374 cosmic_dose_rates_err <- rep(NA,num_samples)
375
376 water_DRAC_alpha_external <- rep(NA,num_samples)
377 water_DRAC_beta_external <- rep(NA,num_samples)
378 water_DRAC_gamma_external <- rep(NA,num_samples)
379 water_DRAC_cosmic <- rep(NA,num_samples)
380 water_DRAC_alpha_internal <- rep(NA,num_samples)
381 water_DRAC_beta_internal <- rep(NA,num_samples)
382
383 for (DRAC_iter in 1:ceiling(num_samples/33)){
384
385   DRAC_start = (DRAC_iter - 1)*33 + 1
386   DRAC_end = min(DRAC_iter*33,num_samples)
387   DRAC_index = DRAC_start:DRAC_end
388   num_DRAC = length(DRAC_index)
389
390   DRAC_table = template_DRAC(nrow=num_DRAC, notification=FALSE)
391
392   DRAC_table$`Project ID` <- rep("USGS-Harrison", num_DRAC)
393   DRAC_table$`Sample ID` <- gsub(pattern=" ",x=sample_list[DRAC_index], replacement = "")
394   DRAC_table$`Mineral` <- mineral_for_DRAC[DRAC_index]
395   DRAC_table$`Conversion factors` <- conversion_factors[DRAC_index]
396
397   DRAC_table$`External U (ppm)` = U_values[DRAC_index]
398   DRAC_table$`errExternal U (ppm)` <- U_err[DRAC_index]
399   DRAC_table$`External Th (ppm)` <- Th_values[DRAC_index]
400   DRAC_table$`errExternal Th (ppm)` <- Th_err[DRAC_index]
401   DRAC_table$`External K (%)` <- K_values[DRAC_index]
402   DRAC_table$`errExternal K (%)` <- K_err[DRAC_index]
403   DRAC_table$`External Rb (ppm)` <- rep("X",num_DRAC)
404   DRAC_table$`errExternal Rb (ppm)` <- rep("X",num_DRAC)
405   DRAC_table$`Calculate external Rb from K conc?` <- calc_Rb_external[DRAC_index]
406
407   DRAC_table$`Internal U (ppm)` <- rep("X",num_DRAC)
408   DRAC_table$`errInternal U (ppm)` <- rep("X",num_DRAC)
409   DRAC_table$`Internal Th (ppm)` <- rep("X",num_DRAC)
410   DRAC_table$`errInternal Th (ppm)` <- rep("X",num_DRAC)
411   DRAC_table$`Internal K (%)` <- internal_K_percent[DRAC_index]
412   DRAC_table$`errInternal K (%)` <- rep(0.5,num_DRAC)
413   DRAC_table$`Rb (ppm)` <- rep("X",num_DRAC)
414   DRAC_table$`errRb (ppm)` <- rep("X",num_DRAC)
415   DRAC_table$`Calculate internal Rb from K conc?` <- calc_Rb_internal[DRAC_index]
416
417   DRAC_table$`User external alphadoserate (Gy.ka-1)` <- rep("X",num_DRAC)
418   DRAC_table$`errUser external alphadoserate (Gy.ka-1)` <- rep("X",num_DRAC)
419   DRAC_table$`User external betadoserate (Gy.ka-1)` <- rep("X",num_DRAC)
420   DRAC_table$`errUser external betadoserate (Gy.ka-1)` <- rep("X",num_DRAC)
421   DRAC_table$`User external gamma doserate (Gy.ka-1)` <- rep("X",num_DRAC)
422   DRAC_table$`errUser external gammadoserate (Gy.ka-1)` <- rep("X",num_DRAC)
423   DRAC_table$`User internal doserate (Gy.ka-1)` <- rep("X",num_DRAC)
424   DRAC_table$`errUser internal doserate (Gy.ka-1)` <- rep("X",num_DRAC)
425   DRAC_table$`User cosmicdoserate (Gy.ka-1)` <- rep("X",num_DRAC)
426   DRAC_table$`errUser cosmicdoserate (Gy.ka-1)` <- rep("X",num_DRAC)
427
428   DRAC_table$`Scale gammadoserate at shallow depths?` <- scale_gamma[DRAC_index]
429   DRAC_table$`Grain size min (microns)` <- grain_size_min[DRAC_index]
430   DRAC_table$`Grain size max (microns)` <- grain_size_max[DRAC_index]
431   DRAC_table$`alpha-Grain size attenuation` <- alpha_attenuation[DRAC_index]
432   DRAC_table$`beta-Grain size attenuation` <- beta_attenuation[DRAC_index]
433   DRAC_table$`Etch depth min (microns)` <- etch_depth_min[DRAC_index]
434   DRAC_table$`Etch depth max (microns)` <- etch_depth_max[DRAC_index]
435   DRAC_table$`beta-Etch depth attenuation factor` <- beta_etch_attenuation_factor[DRAC_index]
436   DRAC_table$`a-value` <- alpha_track[DRAC_index]
437   DRAC_table$`erra-value` <- alpha_track_err[DRAC_index]
438
439   DRAC_table$`Water content ((wet) - dry weight)/dry weight` % <- preferred_water_content_of_samples[DRAC_index]
440   DRAC_table$`errWater content %` <- preferred_water_content_of_samples_err[DRAC_index]
441   DRAC_table$`Depth (m)` <- depth_of_samples[DRAC_index]
442   DRAC_table$`errDepth (m)` <- rep(0.025, num_DRAC)
443   DRAC_table$`Overburden density (g cm-3)` <- rep(180, num_DRAC)
444   DRAC_table$`errOverburden density (g cm-3)` <- rep(0.1, num_DRAC)
445   DRAC_table$`Latitude (decimal degrees)` <- latitude_values[DRAC_index]
446   DRAC_table$`Longitude (decimal degrees)` <- longitude_values[DRAC_index]
447   DRAC_table$`Altitude (m)` <- elevation_values[DRAC_index]

```

```

448 DRAC_table$`De (Gy)` <- rep("X", num_DRAC) # c(152, 71.3, 14.9, 76.7, 67.3) #
449 DRAC_table$`errDe (Gy)` <- rep("X", num_DRAC) # c(8.74, 4.46, 0.71, 4.64, 1.00) #
450
451 DRAC_output = use_DRAC(DRAC_table, name="USGS_Harrison", print_references = FALSE, verbose=FALSE, citation_style="text")
452 DRAC_results <- get_RLum(DRAC_output, "DRAC")
453 write.xlsx(DRAC_results$content, paste("DRAC_output", "-", DRAC_iter, ".xlsx", sep = ""), sheetName="Sheet1")
454
455 dose_rates[DRAC_index] <- as.numeric(DRAC_results$highlights["Environmental Dose Rate (Gy.ka-1)"][1:num_DRAC,1])
456 dose_rates_err[DRAC_index] <- DRAC_results$highlights["errEnvironmental Dose Rate (Gy.ka-1)"][1:num_DRAC,1]
457
458 cosmic_dose_rates[DRAC_index] <- DRAC_results$highlights["Cosmicdose rate (Gy.ka-1)"][1:num_DRAC,1]
459 cosmic_dose_rates_err[DRAC_index] <- DRAC_results$highlights["errCosmicdose rate (Gy.ka-1)"][1:num_DRAC,1]
460
461 water_DRAC_alpha_external[DRAC_index] <- DRAC_results$content["TO:FK"][1:num_DRAC,1]
462 water_DRAC_beta_external[DRAC_index] <- DRAC_results$content["TO:FM"][1:num_DRAC,1]
463 water_DRAC_gamma_external[DRAC_index] <- DRAC_results$content["TO:F0"][1:num_DRAC,1]
464 water_DRAC_cosmic[DRAC_index] <- DRAC_results$content["TO:GG"][1:num_DRAC,1]
465 water_DRAC_alpha_internal[DRAC_index] <- DRAC_results$content["TO:FW"][1:num_DRAC,1]
466 water_DRAC_beta_internal[DRAC_index] <- DRAC_results$content["TO:FY"][1:num_DRAC,1]
467
468 }
469
470 ## prep the final data table for latex
471
472 final_results_table = data.frame(sample_name=sample_list,
473 latitude = latitude_values,
474 longitude = longitude_values,
475 elevation = elevation_values,
476 depths= depth_of_samples,
477 water_content = paste(as.character(round(field_water_content_of_samples)),
478 " (", as.character(round(saturation_water_content_of_samples)),")",
479 " [", as.character(round(preferred_water_content_of_samples)),"]", sep=""),
480 k_percent = paste(sprintf("%.3g", K_values), " $\\pm$ ", sprintf("%.3g", K_err), sep=""),
481 uranium_ppm = paste(sprintf("%.3g", U_values), " $\\pm$ ", sprintf("%.3g", U_err), sep=""),
482 thorium_ppm = paste(sprintf("%.3g", Th_values), " $\\pm$ ", sprintf("%.1g", Th_err), sep=""),
483 cosmic_dose_rate = paste(sprintf("%.3g", cosmic_dose_rates), " $\\pm$ ", sprintf("%.3g", cosmic_
484 dose_rates_err), sep=""),
485 dose_rate = paste(sprintf("%.3g", dose_rates), " $\\pm$ ", sprintf("%.1g", dose_rates_err), sep=""),
486 num_aliquots = rep(NA, num_samples),
487 scatter = rep(NA, num_samples),
488 de_cam = rep(NA, num_samples),
489 de_mam = rep(NA, num_samples),
490 age_cam = rep(NA, num_samples),
491 age_mam = rep(NA, num_samples))
492
493 ## prep the big data table for excel
494
495 final_xlsx_table = data.frame(sample_name = sample_list,
496 latitude = latitude_values,
497 longitude = longitude_values,
498 elevation = elevation_values,
499 sampling_depth= depth_of_samples,
500 field_water_content = field_water_content_of_samples,
501 saturated_water_content = saturation_water_content_of_samples,
502 preferred_water_content_of_samples = preferred_water_content_of_samples,
503 potassium_percent = K_values,
504 potassium_percent_err = K_err,
505 uranium_ppm = U_values,
506 uranium_ppm_err = U_err,
507 thorium_ppm = Th_values,
508 thorium_ppm_err = Th_err,
509 cosmic_dose_rate = cosmic_dose_rates,
510 cosmic_dose_rate_err = cosmic_dose_rates_err,
511 dose_rate = dose_rates,
512 dose_rate_err = dose_rates_err,
513 analysis_type = analysis_type,
514 grain_size_min = grain_size_min,
515 grain_size_max = grain_size_max,
516 num_aliquots_passed = rep(NA, num_samples),
517 num_aliquots_total = rep(NA, num_samples),
518 scatter = rep(NA, num_samples),
519 de_cam = rep(NA, num_samples),
520 de_cam_err = rep(NA, num_samples),
521 de_mam = rep(NA, num_samples),
522 de_mam_err = rep(NA, num_samples),
523 age_cam = rep(NA, num_samples),
524 age_cam_err = rep(NA, num_samples),
525 age_mam = rep(NA, num_samples),
526 age_mam_err = rep(NA, num_samples),
527 age_cam_fade = rep(NA, num_samples),
528 age_cam_fade_err = rep(NA, num_samples),
529 age_mam_fade = rep(NA, num_samples),
530 age_mam_fade_err = rep(NA, num_samples))
531
532 ## prep the miscellaneous parameters tables
533
534 final_misc_table_1 = data.frame(sample_name=sample_list,
535 latitude = latitude_values,
536 longitude = longitude_values,
537 analysis = analysis_type,
538 grains = paste(as.character(grain_size_min), " - ", as.character(grain_size_max), sep=""),

```

```

538         water_type = water_type,
539         DR_method = DR_type)
540
541 final_misc_table_2 = data.frame(sample_name=sample_list,
542                                sigma_b_MAM = sigma_b_input,
543                                MAM_log = MAM_log,
544                                gamma_MAM = rep("NA", num_samples),
545                                sigma_MAM = rep("NA", num_samples),
546                                pO_MAM = rep("NA", num_samples),
547                                mu_MAM = rep("NA", num_samples), stringsAsFactors = FALSE)
548
549 # final_misc_table_2 = data.frame(lapply(final_misc_table_2, gsub, pattern = "NA $\\pm$ NA", replacement = "-", fixed = TRUE),
550                                stringsAsFactors = FALSE)
551
552 final_misc_table_3 = data.frame(sample_name=sample_list,
553                                given_dose = rep("50 Gray"),
554                                Holding_time = hold_time,
555                                g_vals = paste(sprintf("%.3g", g_vals), " $\\pm$ ", sprintf("%.3g", g_value_err), sep=""),
556                                fade_correction_used = fade_correction_used,
557                                fade_correction_type = fade_correction_type)
558
559 ## prep the table for a formatted excel table
560
561 final_summary_table = data.frame(sample_name=sample_list,
562                                latitude = latitude_values,
563                                longitude = longitude_values,
564                                elevation = elevation_values,
565                                depths= depth_of_samples,
566                                water_content = paste(as.character(field_water_content_of_samples),
567                                                       " (", as.character(saturation_water_content_of_samples),")",
568                                                       " [", as.character(round(preferred_water_content_of_samples)),"]", sep=""),
569                                k_percent = paste(sprintf("%.3g", K_values), " ? ", sprintf("%.3g", K_err), sep=""),
570                                uranium_ppm = paste(sprintf("%.3g", U_values), " ? ", sprintf("%.3g", U_err), sep=""),
571                                thorium_ppm = paste(sprintf("%.3g", Th_values), " ? ", sprintf("%.3g", Th_err), sep=""),
572                                cosmic_dose_rate = paste(sprintf("%.3g", cosmic_dose_rates), " ? ", sprintf("%.3g", cosmic_dose_
573                                rates_err), sep=""),
574                                dose_rate = paste(sprintf("%.3g", dose_rates), " ? ", sprintf("%.1g", dose_rates_err), sep=""),
575                                num_aliquots = rep(NA, num_samples),
576                                de_cam = rep(NA, num_samples),
577                                de_mam = rep(NA, num_samples),
578                                age_cam = rep(NA, num_samples),
579                                age_mam = rep(NA, num_samples))
580
581 #####
582
583 ## Build KDE and Radial plots
584
585 sample_iter = 0
586
587 # sample_name = sample_list[sample_iter]
588
589 for (sample_name in sample_list){
590
591 sample_iter = sample_iter + 1
592 print(paste("Running sample: ", sample_list[sample_iter], sep=""))
593
594 if (bin_or_xlsx[sample_iter] == 'bin'){
595
596     bin_data = read_BIN2R(file = file_names[sample_iter],
597                           show.raw.values = FALSE,
598                           position = NULL,
599                           n.records = NULL,
600                           zero_data.rm = TRUE,
601                           duplicated.rm = TRUE,
602                           fastForward = FALSE,
603                           show.record.number = FALSE,
604                           txtProgressBar = TRUE,
605                           forced.VersionNumber = NULL,
606                           ignore.RECTYPE = 1,
607                           pattern = NULL,
608                           verbose = TRUE)
609
610     RLum_object = Risoe.BINfileData2RLum.Analysis(bin_data,
611                                                    pos = NULL,
612                                                    grain = NULL,
613                                                    run = NULL,
614                                                    set = NULL,
615                                                    ltype = OSL_IRSL_type[sample_iter],
616                                                    dtype = NULL,
617                                                    protocol = "unknown",
618                                                    keep.empty = FALSE,
619                                                    txtProgressBar = FALSE)
620
621     if ((OSL_IRSL_type[sample_iter]=="OSL") & (any(bin_data@METADATA$LTYPE=="IRSL"))){
622
623         RLum_object_IRSL = Risoe.BINfileData2RLum.Analysis(bin_data,
624                                                             pos = NULL,
625                                                             grain = NULL,
626                                                             run = NULL,
627                                                             set = NULL,

```

```

628                                     ltype = "IRSL",
629                                     dtype = NULL,
630                                     protocol = "unknown",
631                                     keep.empty = FALSE,
632                                     txtProgressBar = FALSE)
633   } else {
634     print("No IRSL depletion test detected")
635     RLum_object_IRSL = NULL
636   }
637
638   meas_date <- as.Date(paste("20",
639                               substr(bin_data@METADATA$DATE[1],5,6),
640                               "-",
641                               substr(bin_data@METADATA$DATE[1],3,4),
642                               "-",
643                               substr(bin_data@METADATA$DATE[1],1,2),sep=""))
644
645   if (machine[sample_iter] == 'LR'){
646
647     machine_dose_rate = calc_SourceDoseRate(measurement.date = meas_date,
648                                             calib.date=as.Date("2020-01-29 00:00:00 MST"),
649                                             calib.dose.rate=0.106609808,
650                                             calib.error=0.000909252,
651                                             source.type = "Sr-90",
652                                             dose.rate.unit = "Gy/s",
653                                             predict = NULL)
654
655     print(paste('Using Dose Rate for Lexsyg Research:',round(machine_dose_rate$dose.rate$dose.rate,4),"+/-",round(machine_dose_
656                                                           rate$dose.rate$dose.rate.error,4)))
657   }
658
659   if (machine[sample_iter] == 'LS'){
660
661     machine_dose_rate = calc_SourceDoseRate(measurement.date = meas_date,
662                                             calib.date=as.Date("2019-09-10 00:00:00 MST"),
663                                             calib.dose.rate=0.111607143,
664                                             calib.error=0.002989477,
665                                             source.type = "Sr-90",
666                                             dose.rate.unit = "Gy/s",
667                                             predict = NULL)
668
669     print(paste('Using Dose Rate for Lexsyg Smart:',round(machine_dose_rate$dose.rate$dose.rate,4),"+/-",round(machine_dose_rate$
670                                                           dose.rate$dose.rate.error,4)))
671   }
672
673   if (machine[sample_iter] == 'R387'){
674
675     machine_dose_rate = calc_SourceDoseRate(measurement.date = meas_date,
676                                             calib.date=as.Date("2015-06-16 00:00:00 MST"),
677                                             calib.dose.rate=0.0914,
678                                             calib.error=0.0028,
679                                             source.type = "Sr-90",
680                                             dose.rate.unit = "Gy/s",
681                                             predict = NULL)
682
683     print(paste('Using Dose Rate for Risoe 387:',round(machine_dose_rate$dose.rate$dose.rate,4),"+/-",round(machine_dose_rate$dose
684                                                           .rate$dose.rate.error,4)))
685   }
686
687   if (machine[sample_iter] == 'R96'){
688
689     machine_dose_rate = calc_SourceDoseRate(measurement.date = meas_date,
690                                             calib.date=as.Date("2015-06-16 00:00:00 MST"),
691                                             calib.dose.rate=0.0707,
692                                             calib.error=0.0038,
693                                             source.type = "Sr-90",
694                                             dose.rate.unit = "Gy/s",
695                                             predict = NULL)
696
697     print(paste('Using Dose Rate for Risoe 96:',round(machine_dose_rate$dose.rate$dose.rate,4),"+/-",round(machine_dose_rate$dose
698                                                           .rate$dose.rate.error,4)))
699   }
700
701   results <- analyse_SAR.CWOSL(
702     object = RLum_object,
703     signal.integral.min = sig_min[sample_iter],
704     signal.integral.max = sig_max[sample_iter],
705     background.integral.min = background_min[sample_iter],
706     background.integral.max = background_max[sample_iter],
707     fit.method = fit_eqn[sample_iter],
708     rejection.criteria = list(
709       recycling.ratio = Recyc_Ratio[sample_iter],
710       recuperation.rate = Recup_values[sample_iter],
711       testdose.error = TestErr[sample_iter],
712       palaeodose.error = PaleoErr[sample_iter],
713       exceed.max.regpoint = ExceedMax[sample_iter]),
714     plot=FALSE,
715     plot.single = FALSE,
716     verbose=FALSE
717     # onlyLxTxTable = TRUE

```

```

716 )
717
718
719 # Wash the outputs and convert to Grays
720
721 results_subset = subset(results$data, RC.Status == "OK")
722
723 if (nrow(results_subset)<3){
724   print(paste("Not enough aliquots passing criteria for sample ", sample_name,sep=""))
725   next
726 }
727
728 write.xlsx(results_subset, paste(sample_name,"Equivalent Dose Spreadsheet.xlsx",sep=" "),sheetName=sample_name)
729 write.xlsx(results$LnLxTx.table, paste(sample_name,"LnLxTx Spreadsheet.xlsx",sep=" "),sheetName=sample_name)
730
731 file.move(paste(wdir,sample_name," Equivalent Dose Spreadsheet.xlsx",sep=""), paste(wdir,"output_tables/" ,sep=""),overwrite=
TRUE)
732 file.move(paste(wdir,sample_name," LnLxTx Spreadsheet.xlsx",sep=""), paste(wdir,"output_tables/" ,sep=""),overwrite=TRUE)
733
734 plot_input = Second2Gray(data=results_subset[c("De","De.Error")],
735   dose.rate = machine_dose_rate,
736   error.propagation = "absolute")
737
738 # Convert to Gray
739 # number of accepted aliquots
740
741 num_accepted[sample_iter] = nrow(subset(results$data, RC.Status == "OK"))
742 num_total[sample_iter] = nrow(subset(results$data))
743 print(paste(toString(num_accepted[sample_iter]),
744   ' aliquots out of ',
745   toString(num_total[sample_iter]),
746   ' accepted.',sep=""))
747
748 if (bin_or_xlsx[sample_iter] == 'xlsx'){
749
750   plot_input = read.xlsx(file_names[sample_iter], colNames = TRUE, rowNames = FALSE, skipEmptyRows = TRUE)
751   num_accepted[sample_iter] = nrow(plot_input)
752   num_total[sample_iter] = "48"
753
754 }
755
756 if (bin_or_xlsx[sample_iter] == 'NA'){
757   plot_input = data.frame(De=rep(NA,10),De.Error=rep(NA,10)) # dummy data
758 }
759
760 # Calculate CAM and MAM Equivalent Doses
761
762 central_age = calc_CentralDose(plot_input, log = TRUE, plot = FALSE) #sigmab=0.15,
763 minimum_age = calc_MinDose(plot_input,
764   sigmab=sigma_b_input[sample_iter],
765   log = MAM_log[sample_iter], par = 3, bootstrap = bootstrap_multi_core_mam_true_or_false,
766   level = 0.95, plot = FALSE, multicore = bootstrap_multi_core_mam_true_or_false) #
767
768 CA = as.numeric(central_age["summary"]["de"])
769 CA_err = as.numeric(central_age["summary"]["de_err"])
770 MA = as.numeric(minimum_age["summary"]["de"])
771 MA_err = as.numeric(minimum_age["summary"]["de_err"])
772
773 if (use_FMM[sample_iter] == TRUE){
774   FMM = calc_FiniteMixture(data= plot_input,
775     sigmab = sigma_b_input,
776     n.components = 2,
777     grain.probability = FALSE,
778     # dose.scale,
779     # pdf.weight = TRUE,
780     # pdf.sigma = "sigmab",
781     # pdf.colors = "gray",
782     # pdf.scale,
783     # plot.proportions = TRUE,
784     plot = TRUE
785   )
786
787   CA = FMM$summary$de[2]
788   CA_err = FMM$summary$de_err[2]
789   MA = FMM$summary$de[1]
790   MA_err = FMM$summary$de_err[1]
791
792 }
793
794 CA_age = CA/dose_rates[sample_iter]
795 MA_age = MA/dose_rates[sample_iter]
796
797 # calculate age uncertainty
798 CA_age_uncert = (CA/dose_rates[sample_iter])*sqrt(((CA_err/CA)^2 + (dose_rates_err[sample_iter]/dose_rates[sample_iter])^2)
799 MA_age_uncert = (MA/dose_rates[sample_iter])*sqrt(((MA_err/MA)^2 + (dose_rates_err[sample_iter]/dose_rates[sample_iter])^2)
800
801 if (OSL_IRSL_type[sample_iter]=="IRSL" || OSL_IRSL_type[sample_iter]=="pIRSL"){
802
803   if (fade_correction_type[sample_iter] == "Huntley and Lamothe, 2001"){
804
805     print(paste("Performing Huntley and Lamothe 2001 fading Correction for sample: ", sample_name,sep=""))
806

```

```

807 | if (fade_correction_used[sample_iter]==TRUE){
808 |
809 |   g_val_input = c(g_vals[sample_iter], g_value_err[sample_iter])
810 |
811 |   CA_fadeCorr <- calc_FadingCorr(age.faded=c(CA_age,CA_age_uncert),
812 |     g_value = g_val_input,
813 |     tc=tc_val[sample_iter],
814 |     tc.g_value = 2*24*60*60,
815 |     txtProgressBar = FALSE,
816 |     verbose = FALSE)
817 |
818 |
819 |   CA_age_fade = as.numeric(CA_fadeCorr$age.corr["AGE"][1])
820 |   CA_age_fade_uncert = as.numeric(CA_fadeCorr$age.corr["AGE.ERROR"][1])
821 |
822 |   MA_fadeCorr <- calc_FadingCorr(age.faded=c(MA_age,MA_age_uncert),
823 |     g_value = g_val_input,
824 |     tc=tc_val[sample_iter],
825 |     tc.g_value = 2*24*60*60,
826 |     txtProgressBar = FALSE,
827 |     verbose = FALSE)
828 |
829 |   MA_age_fade = as.numeric(MA_fadeCorr$age.corr["AGE"][1])
830 |   MA_age_fade_uncert = as.numeric(MA_fadeCorr$age.corr["AGE.ERROR"][1])
831 |
832 |
833 | } else {
834 |
835 |   print("skipping fade correction")
836 |
837 |   CA_age_fade = CA_age
838 |   CA_age_fade_uncert = CA_age_uncert
839 |
840 |   MA_age_fade = MA_age
841 |   MA_age_fade_uncert = MA_age_uncert
842 |
843 | }
844 |
845 |
846 |
847 | # if (pigati_prefs==TRUE){
848 | #
849 | #   CA_faded = CA_age*dose_rates[sample_iter]
850 | #   CA_faded_err = CA*sqrt( (CA_age_uncert/CA_age)^2 + (dose_rates_err[sample_iter]/dose_rates[sample_iter])^2)
851 | #
852 | #   print("CA faded DEs")
853 | #   print(CA_faded)
854 | #   print(CA_faded_err)
855 | #
856 | #   CA = CA_faded
857 | #   CA_err = CA_faded_err
858 | #
859 | #   MA_faded = MA_age*dose_rates[sample_iter]
860 | #   MA_faded_err = MA*sqrt( (MA_age_uncert/MA_age)^2 + (dose_rates_err[sample_iter]/dose_rates[sample_iter])^2)
861 | #
862 | #   print("MA faded DEs")
863 | #   print(MA_faded)
864 | #   print(MA_faded_err)
865 | #
866 | #   MA = MA_faded
867 | #   MA_err = MA_faded_err
868 | #
869 | # }
870 |
871 |
872 |
873 | }
874 |
875 | if (fade_correction_type[sample_iter] == "huntley2006"){
876 |
877 |   print(paste("Performing Huntley 2006 fading Correction for sample: ", sample_name,sep=""))
878 |
879 |   CA_fadeCorr <- calc_Huntley2006(data, LnTn = NULL, rhop, ddot, readerDdot,
880 |     normalise = TRUE, fit.method = c("EXP", "GOK")[1],
881 |     lower.bounds = c(-Inf, -Inf, -Inf), summary = TRUE, plot = TRUE,
882 |     ...)
883 |
884 | }
885 |
886 |
887 | }
888 |
889 | final_results_table["scatter"][sample_iter,1] = paste(round(central_age$summary["OD"],0), "%",sep="")
890 | final_xlsx_table["scatter"][sample_iter,1] = paste(round(central_age$summary["OD"],0), "%",sep="")
891 |
892 | ## Input results into summary tables ##
893 |
894 |
895 | final_results_table["num_aliquots"][sample_iter,1] = paste(toString(num_accepted[sample_iter]),
896 |   " (",
897 |   toString(num_total[sample_iter]),
898 |   ")",

```

```

899         sep="")
900
901 final_summary_table["num_aliquots"][sample_iter,1] = paste(toString(num_accepted[sample_iter]),
902     " (",
903     toString(num_total[sample_iter]),
904     ")",
905     sep="")
906 # enter in mam and cam De
907 final_results_table["de_mam"][sample_iter,1] = paste(formatC(signif(MA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag="#"),
908     " $\\pm$ ",
909     formatC(signif(MA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag=
910     "#"), sep="")
911 final_results_table["de_cam"][sample_iter,1] = paste(formatC(signif(CA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag="#"),
912     " $\\pm$ ",
913     formatC(signif(CA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag=
914     "#"), sep="")
915 final_summary_table["de_mam"][sample_iter,1] = paste(formatC(signif(MA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag="#"),
916     " ? ",
917     formatC(signif(MA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag=
918     "#"), sep="")
919 final_summary_table["de_cam"][sample_iter,1] = paste(formatC(signif(CA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag="#"),
920     " ? ",
921     formatC(signif(CA_err,3), format = "f", digits= 1,drop0trailing = FALSE, flag=
922     "#"), sep="")
923 # enter in ages and uncertainties
924 if (OSL_IRSL_type[sample_iter]=="IRSL"){ # input fading corrected ages
925     final_results_table["age_cam"][sample_iter,1] = paste(sprintf("%.3g",CA_age_fade)," $\\pm$ ",
926         formatC(signif(CA_age_fade_uncert,3), format = "f", digits= 1,
927         drop0trailing = FALSE, flag="#"), sep="")
928     final_results_table["age_mam"][sample_iter,1] = paste(sprintf("%.3g",MA_age)," $\\pm$ ",
929         formatC(signif(MA_age_uncert,3), format = "f", digits= 1,drop0trailing =
930         FALSE, flag="#"), sep="")
931     final_summary_table["age_cam"][sample_iter,1] = paste(sprintf("%.3g",CA_age_fade)," ? ",
932         formatC(signif(CA_age_fade_uncert,3), format = "f", digits= 1,
933         drop0trailing = FALSE, flag="#"), sep="")
934     final_summary_table["age_mam"][sample_iter,1] = paste(sprintf("%.3g",MA_age_fade)," ? ",
935         formatC(signif(MA_age_fade_uncert,3), format = "f", digits= 1,
936         drop0trailing = FALSE, flag="#"), sep="")
937 } else {
938     final_results_table["age_cam"][sample_iter,1] = paste(sprintf("%.3g",CA_age)," $\\pm$ ",
939         formatC(signif(CA_age_uncert,3), format = "f", digits= 1,drop0trailing =
940         FALSE, flag="#"), sep="")
941     final_results_table["age_mam"][sample_iter,1] = paste(sprintf("%.3g",MA_age)," $\\pm$ ",
942         formatC(signif(MA_age_uncert,3), format = "f", digits= 1,drop0trailing =
943         FALSE, flag="#"), sep="")
944     final_summary_table["age_cam"][sample_iter,1] = paste(sprintf("%.3g",CA_age)," ? ",
945         formatC(signif(CA_age_uncert,3), format = "f", digits= 1,drop0trailing =
946         FALSE, flag="#"), sep="")
947     final_summary_table["age_mam"][sample_iter,1] = paste(sprintf("%.3g",MA_age)," ? ",
948         formatC(signif(MA_age_uncert,3), format = "f", digits= 1,drop0trailing =
949         FALSE, flag="#"), sep="")
950 }
951 # print(formatC(signif(MA_err,digits=3), digits=2,format="fg", flag="#"))
952 # add to final excel table
953 final_xlsx_table["num_aliquots_passed"][sample_iter,1] <- num_accepted[sample_iter]
954 final_xlsx_table["num_aliquots_total"][sample_iter,1] <- num_total[sample_iter]
955 final_xlsx_table["de_cam"][sample_iter,1] <- CA
956 final_xlsx_table["de_cam_err"][sample_iter,1] <- CA_err
957 final_xlsx_table["de_mam"][sample_iter,1] <- MA
958 final_xlsx_table["de_mam_err"][sample_iter,1] <- MA_err
959 final_xlsx_table["age_cam"][sample_iter,1] <- CA_age
960 final_xlsx_table["age_cam_err"][sample_iter,1] <- CA_age_uncert
961 final_xlsx_table["age_mam"][sample_iter,1] <- MA_age
962 final_xlsx_table["age_mam_err"][sample_iter,1] <- MA_age_uncert
963 if (OSL_IRSL_type[sample_iter]=="IRSL"){
964     final_xlsx_table["age_cam_fade"][sample_iter,1] <- CA_age_fade

```

```

975 final_xlsx_table["age_cam_fade_err"][sample_iter,1] <- CA_age_fade_uncert
976
977 final_xlsx_table["age_mam_fade"][sample_iter,1] <- MA_age_fade
978 final_xlsx_table["age_mam_fade_err"][sample_iter,1] <- MA_age_fade_uncert
979
980 } else {
981
982   final_xlsx_table["age_cam_fade"][sample_iter,1] <- "-"
983   final_xlsx_table["age_cam_fade_err"][sample_iter,1] <- "-"
984
985   final_xlsx_table["age_mam_fade"][sample_iter,1] <- "-"
986   final_xlsx_table["age_mam_fade_err"][sample_iter,1] <- "-"
987
988 }
989
990
991 # add to misc table
992
993 final_misc_table_2[c("gamma_MAM", "sigma_MAM", "p0_MAM", "mu_MAM")][sample_iter,] =
994   c(round(as.numeric(unlist(minimum_age$mle@coef["gamma"])),3),
995     round(as.numeric(unlist(minimum_age$mle@coef["sigma"])),3),
996     round(as.numeric(unlist(minimum_age$mle@coef["p0"])),3),
997     round(as.numeric(unlist(minimum_age$mle@coef["mu"])),3))
998
999 #
1000 # final_misc_table_3[c("age_cam", "age_cam_fade", "age_mam", "age_mam_fade")][sample_iter,] =
1001 #   c(paste(sprintf("%.3g",CA_age)," $\\pm$",formatC(signif(CA_age_uncert,3), format = "f", digits= 1,drop0trailing = FALSE, flag
1002 #     = "#"), sep=""),
1003 #     paste(sprintf("%.3g",CA_age_fade)," $\\pm$", formatC(signif(CA_age_fade_uncert,3), format = "f", digits= 1,drop0trailing =
1004 #     FALSE, flag="#"), sep=""),
1005 #     paste(sprintf("%.3g",MA_age)," $\\pm$", formatC(signif(MA_age_uncert,3), format = "f", digits= 1,drop0trailing = FALSE, flag
1006 #     = "#"), sep=""),
1007 #     paste(sprintf("%.3g",MA_age_fade)," $\\pm$", formatC(signif(MA_age_fade_uncert,3), format = "f", digits= 1,drop0trailing =
1008 #     FALSE, flag="#"), sep=""))
1009
1010 #####
1011 ## Determine the sensitivity of the water content assumption
1012
1013 print("Performing Water Content Sensitivity Analysis")
1014
1015 water_array = seq(0,1.2*saturation_water_content_of_samples[sample_iter],1)/100
1016
1017 x_alpha = 1.49
1018 x_beta = 1.25
1019 x_gamma = 1.14
1020
1021 water_dose_rates = (water_DRAC_alpha_external[sample_iter]/(1 + x_alpha*water_array) + # external alpha dose
1022   water_DRAC_beta_external[sample_iter]/(1 + x_beta*water_array) + # external beta dose
1023   water_DRAC_gamma_external[sample_iter]/(1 + x_gamma*water_array) + # external gamma dose
1024   water_DRAC_cosmic[sample_iter] + # cosmic dose
1025   water_DRAC_alpha_internal[sample_iter] + # internal alpha dose
1026   water_DRAC_beta_internal[sample_iter]) # internal beta dose
1027
1028 if (OSL_IRSL_type[sample_iter]=="IRSL"){ # to account for fading
1029
1030   print("creating IRSL water plot")
1031
1032   # water_CA_age = CA_age_fade
1033   # water_CA_age_uncert = CA_age_fade_uncert
1034   #
1035   # water_MA_age = MA_age_fade
1036   # water_MA_age_uncert = MA_age_fade_uncert
1037   #
1038   # water_CAM_array = (water_CA_age*dose_rates[sample_iter])/water_dose_rates
1039   # water_MAM_array = (water_MA_age*dose_rates[sample_iter])/water_dose_rates
1040   #
1041   # water_data_frame = data.frame(water=100*rep(water_array,2),
1042   #   age=c(water_CAM_array),
1043   #   age_uncert_pos = c(water_CAM_array + water_CA_age_uncert),
1044   #   age_uncert_neg = c(water_CAM_array - water_CA_age_uncert),
1045   #   age_type=c(rep("CAM Age",length(water_CAM_array))))
1046   #
1047   # p_water <- ggplot(data=water_data_frame,aes(x=water,y=age,group=age_type)) +
1048   #   scale_color_manual(values=c("red","blue"), name = "Age Model") +
1049   #   scale_fill_manual(values=c("red","blue"), name="95% Conf. Int.") +
1050   #   geom_line(aes(color=age_type),size = 1.25) +
1051   #   geom_ribbon(aes(ymax = age_uncert_pos,
1052   #     ymin=age_uncert_neg,
1053   #     x=water,
1054   #     fill=age_type),
1055   #     alpha = .1) +
1056   #   geom_vline(xintercept = c(field_water_content_of_samples[sample_iter],
1057   #     field_water_content_of_samples[sample_iter]*1.05,
1058   #     field_water_content_of_samples[sample_iter]*0.95,
1059   #     saturation_water_content_of_samples[sample_iter],
1060   #     saturation_water_content_of_samples[sample_iter]*1.05,
1061   #     saturation_water_content_of_samples[sample_iter]*0.95),
1062   #     # preferred_water_content_of_samples[sample_iter],
1063   #     # preferred_water_content_of_samples[sample_iter]+preferred_water_content_of_samples_err[sample_
1064   #     iter],

```



```

1062 #           # preferred_water_content_of_samples[sample_iter]-preferred_water_content_of_samples_err[sample_
1063 iter]],
1064 #           size=0.5,
1065 #           linetype=c("solid","dashed","dashed",
1066 # "solid","dashed","dashed")) +
1067 #           # "solid","dashed","dashed")) +
1068 # geom_errorbar(aes(x = preferred_water_content_of_samples[sample_iter],
1069 # ymin = water_CA_age - water_CA_age_uncert,
1070 # ymax = water_CA_age + water_CA_age_uncert), size = 0.25, width = 0.75) +
1071 # geom_point(aes(x = preferred_water_content_of_samples[sample_iter],
1072 # y = water_CA_age), shape = 21, colour = "black", fill = "red", size = 3, stroke = 0.5) +
1073 # geom_text(aes(x=saturation_water_content_of_samples[sample_iter]*1.05,
1074 # y=median(water_CAM_array),
1075 # label = "Saturation Moisture",
1076 # angle=90,
1077 # vjust=1.3)) +
1078 # geom_text(aes(x=field_water_content_of_samples[sample_iter]*1.05,
1079 # y=median(water_CAM_array),
1080 # label = "Field Moisture",
1081 # angle=90,
1082 # vjust=1.3)) +
1083 # # geom_text(aes(x=preferred_water_content_of_samples[sample_iter] - 1.5,
1084 # # y=median(water_CAM_array),
1085 # # label = "Assumed Water Content",
1086 # # angle=90)) +
1087 # theme_bw() +
1088 # labs(title = "D) Sensitivity of Age to Percent Water Content", #title =paste("D) Sensitivity of Age to Percent Water Content
1089 # for ",sample_name,sep="")
1090 # x="Percent Water by Weight", y="Age (ka)"
1091
1092 water_CA_age = CA_age
1093 water_CA_age_uncert = CA_age_uncert
1094
1095 water_MA_age = MA_age
1096 water_MA_age_uncert = MA_age_uncert
1097
1098 water_CAM_array = (water_CA_age*dose_rates[sample_iter])/water_dose_rates
1099 water_MAM_array = (water_MA_age*dose_rates[sample_iter])/water_dose_rates
1100
1101 if (any(is.na(c(water_array, water_CAM_array, water_MAM_array)))){
1102   stop("NA's in the water array!")
1103 }
1104
1105 water_data_frame = data.frame(water=100*rep(water_array,2),
1106                               age=c(water_CAM_array,water_MAM_array),
1107                               age_uncert_pos = c(water_CAM_array + water_CA_age_uncert, water_MAM_array + water_MA_age_uncert),
1108                               age_uncert_neg = c(water_CAM_array - water_CA_age_uncert, water_MAM_array - water_MA_age_uncert),
1109                               age_type=c(rep("CAM Age",length(water_CAM_array)),
1110                                           rep("MAM Age",length(water_MAM_array))))
1111
1112 ## Plot the water chart
1113 p_water <- ggplot(data=water_data_frame,aes(x=water,y=age,group=age_type)) +
1114   scale_color_manual(values=c("red","blue"), name = "Age Model") +
1115   scale_fill_manual(values=c("red","blue"), name="95% Conf. Int.") +
1116   geom_line(aes(color=age_type),size = 1.25) +
1117   geom_ribbon(aes(ymax = age_uncert_pos,
1118                 ymin=age_uncert_neg,
1119                 x=water,
1120                 fill=age_type),
1121             alpha = .1) +
1122   geom_vline(xintercept = c(field_water_content_of_samples[sample_iter],
1123                             field_water_content_of_samples[sample_iter]*1.05,
1124                             field_water_content_of_samples[sample_iter]*0.95,
1125                             saturation_water_content_of_samples[sample_iter],
1126                             saturation_water_content_of_samples[sample_iter]*1.05,
1127                             saturation_water_content_of_samples[sample_iter]*0.95),
1128             # preferred_water_content_of_samples[sample_iter],
1129             # preferred_water_content_of_samples[sample_iter]+preferred_water_content_of_samples_err[sample_iter],
1130             # preferred_water_content_of_samples[sample_iter]-preferred_water_content_of_samples_err[sample_iter]),
1131             size=0.5,
1132             linetype=c("solid","dashed","dashed",
1133                       "solid","dashed","dashed")) +
1134   # "solid","dashed","dashed")) +
1135   geom_errorbar(aes(x = preferred_water_content_of_samples[sample_iter],
1136                     ymin = water_CA_age - water_CA_age_uncert,
1137                     ymax = water_CA_age + water_CA_age_uncert), size = 0.25, width = 0.75) +
1138   geom_point(aes(x = preferred_water_content_of_samples[sample_iter],
1139                 y = water_CA_age), shape = 21, colour = "black", fill = "red", size = 3, stroke = 0.5) +
1140   geom_errorbar(aes(x = preferred_water_content_of_samples[sample_iter],
1141                     ymin = water_MA_age - water_MA_age_uncert,
1142                     ymax = water_MA_age + water_MA_age_uncert), size = 0.25, width = 0.75) +
1143   geom_point(aes(x = preferred_water_content_of_samples[sample_iter],
1144                 y = water_MA_age), shape = 21, colour = "black", fill = "blue", size = 3, stroke = 0.5) +
1145   geom_text(aes(x=saturation_water_content_of_samples[sample_iter]*1.05,
1146                 y=median(water_CAM_array),
1147                 label = "Saturation Moisture",
1148                 angle=90,
1149                 vjust=1.3)) +
1150   geom_text(aes(x=field_water_content_of_samples[sample_iter]*1.05,
1151                 y=median(water_CAM_array),
1152                 label = "Field Moisture",
1153                 angle=90,
1154                 vjust=1.3)) +

```

```

1152 # geom_text(aes(x=preferred_water_content_of_samples[sample_iter]*1.05,
1153 #               y=median(water_CAM_array),
1154 #               label = "Modeled Moisture",
1155 #               angle=90,
1156 #               vjust=1.3)) +
1157 theme_bw() +
1158 labs(title = "D) Sensitivity of Age to Percent Water Content", #title =paste("D) Sensitivity of Age to Percent Water Content
1159       for " ,sample_name,sep="")
1160       x="Percent Water by Weight", y="Age (ka)")
1161
1162 } else {
1163
1164   print("creating OSL water plot")
1165
1166   water_CA_age = CA_age
1167   water_CA_age_uncert = CA_age_uncert
1168
1169   water_MA_age = MA_age
1170   water_MA_age_uncert = MA_age_uncert
1171
1172   water_CAM_array = (water_CA_age*dose_rates[sample_iter])/water_dose_rates
1173   water_MAM_array = (water_MA_age*dose_rates[sample_iter])/water_dose_rates
1174
1175   if (any(is.na(c(water_array, water_CAM_array, water_MAM_array)))){
1176     stop("NA's in the water array!")
1177   }
1178
1179   water_data_frame = data.frame(water=100*rep(water_array,2),
1180                                 age=c(water_CAM_array,water_MAM_array),
1181                                 age_uncert_pos = c(water_CAM_array + water_CA_age_uncert, water_MAM_array + water_MA_age_uncert),
1182                                 age_uncert_neg = c(water_CAM_array - water_CA_age_uncert, water_MAM_array - water_MA_age_uncert),
1183                                 age_type=c(rep("CAM Age",length(water_CAM_array)),
1184                                             rep("MAM Age",length(water_MAM_array))))
1185
1186   ## Plot the water chart
1187   p_water <- ggplot(data=water_data_frame,aes(x=water,y=age,group=age_type)) +
1188     scale_color_manual(values=c("red","blue"), name = "Age Model") +
1189     scale_fill_manual(values=c("red","blue"), name="95% Conf. Int.") +
1190     geom_line(aes(color=age_type),size = 1.25) +
1191     geom_ribbon(aes(ymin = age_uncert_pos,
1192                   ymin=age_uncert_neg,
1193                   x=water,
1194                   fill=age_type),
1195               alpha = .1) +
1196     geom_vline(xintercept = c(field_water_content_of_samples[sample_iter],
1197                               field_water_content_of_samples[sample_iter]*1.05,
1198                               field_water_content_of_samples[sample_iter]*0.95,
1199                               saturation_water_content_of_samples[sample_iter],
1200                               saturation_water_content_of_samples[sample_iter]*1.05,
1201                               saturation_water_content_of_samples[sample_iter]*0.95),
1202               # preferred_water_content_of_samples[sample_iter],
1203               # preferred_water_content_of_samples[sample_iter]+preferred_water_content_of_samples_err[sample_iter
1204               ],
1205               # preferred_water_content_of_samples[sample_iter]-preferred_water_content_of_samples_err[sample_iter
1206               ]),
1207               size=0.5,
1208               linetype=c("solid","dashed","dashed",
1209                           "solid","dashed","dashed")) +
1210     # "solid","dashed","dashed")) +
1211     geom_errorbar(aes(x = preferred_water_content_of_samples[sample_iter],
1212                      ymin = water_CA_age - water_CA_age_uncert,
1213                      ymax = water_CA_age + water_CA_age_uncert), size = 0.25, width = 0.75) +
1214     geom_point(aes(x = preferred_water_content_of_samples[sample_iter],
1215                   y = water_CA_age), shape = 21, colour = "black", fill = "red", size = 3, stroke = 0.5) +
1216     geom_errorbar(aes(x = preferred_water_content_of_samples[sample_iter],
1217                      ymin = water_MA_age - water_MA_age_uncert,
1218                      ymax = water_MA_age + water_MA_age_uncert), size = 0.25, width = 0.75) +
1219     geom_point(aes(x = preferred_water_content_of_samples[sample_iter],
1220                   y = water_MA_age), shape = 21, colour = "black", fill = "blue", size = 3, stroke = 0.5) +
1221     geom_text(aes(x=saturation_water_content_of_samples[sample_iter]*1.05,
1222                  y=median(water_CAM_array),
1223                  label = "Saturation Moisture",
1224                  angle=90,
1225                  vjust=1.3)) +
1226     geom_text(aes(x=field_water_content_of_samples[sample_iter]*1.05,
1227                  y=median(water_CAM_array),
1228                  label = "Field Moisture",
1229                  angle=90,
1230                  vjust=1.3)) +
1231     # geom_text(aes(x=preferred_water_content_of_samples[sample_iter]*1.05,
1232     #               y=median(water_CAM_array),
1233     #               label = "Modeled Moisture",
1234     #               angle=90,
1235     #               vjust=1.3)) +
1236     theme_bw() +
1237     labs(title = "D) Sensitivity of Age to Percent Water Content", #title =paste("D) Sensitivity of Age to Percent Water Content
1238           for " ,sample_name,sep="")
1239           x="Percent Water by Weight", y="Age (ka)")
1240
1241 }
1242
1243 #####

```

```

1240
1241 # Plot the results of the aliquots as PDF
1242 # pdf(file = paste(wdir,"/output_plots/",sample_name,"_plots.pdf",sep=""),
1243 #     paper = "letter", height=10)
1244 # par(mfrow = c(2, 1))
1245 #
1246 # plot_KDE(plot_input,
1247 #           main=paste(sample_name,": ", "De Distribution (Kernel Density Estimate)",sep=""),
1248 #           summary = c("n","median","kurtosis","skewness"),
1249 #           xlab = "Equivalent dose [Gy]")
1250 #
1251 # abline(v=CA, col="red")
1252 # abline(v=CA-CA_err, col="red",lty=2)
1253 # abline(v=CA+CA_err, col="red",lty=2)
1254 #
1255 # abline(v=MA, col="blue")
1256 # abline(v=MA-MA_err, col="blue",lty=2)
1257 # abline(v=MA+MA_err, col="blue",lty=2)
1258 #
1259 #
1260 # plot_RadialPlot(plot_input,
1261 #                 main=paste(sample_name,": ", "Radial Plot",sep=""),
1262 #                 centrality = "mean.weighted",
1263 #                 log.z = TRUE,
1264 #                 line = c(CA, MA),
1265 #                 line.col = c("red","blue"),
1266 #                 summary = c("n","mean.weighted","seabs.weighted","in.2s"),
1267 #                 summary.pos = "topleft",
1268 #                 grid.col = "none",
1269 #                 pch=16
1270 #                 )
1271 #
1272 #
1273 #
1274 #
1275 # plot_RadialPlot(data = plot_input,
1276 #                 na.rm = TRUE,
1277 #                 log.z = TRUE,
1278 #                 xlim = c(0,50),
1279 #                 # central.value,
1280 #                 centrality = "mean.weighted",
1281 #                 main=paste(sample_name,": ", "Radial Plot",sep=""),
1282 #                 # mtext = sample_name,
1283 #                 summary = c("n","mean.weighted","seabs.weighted","in.2s"),
1284 #                 # summary.pos = "topleft",
1285 #                 # legend,
1286 #                 # legend.pos,
1287 #                 # stats,
1288 #                 rug = FALSE,
1289 #                 # plot.ratio = 10,
1290 #                 # bar.col,
1291 #                 y.ticks = TRUE,
1292 #                 grid.col = "none",
1293 #                 line = c(CA, MA),
1294 #                 line.col = c("red","blue"),
1295 #                 # line.label,
1296 #                 # output = FALSE,
1297 #                 pch=16)
1298 # dev.off()
1299
1300 #####
1301
1302 ## Plot the results of the aliquots as PNG
1303 png(file = paste(wdir, sample_name,"_plots.png",sep=""),
1304     width = 1000, height = 500)
1305 par(mfrow = c(1, 2))
1306
1307 if (OSL_IRSL_type[sample_iter]=="OSL"){
1308
1309   p_KDE <- plot_KDE(plot_input,
1310                     main=paste(sample_name,": ", "De Distribution (Kernel Density Estimate)",sep=""),
1311                     summary = c("n","median","kurtosis","skewness"),
1312                     xlab = "Equivalent dose [Gy]",
1313                     output = FALSE)
1314
1315   abline(v=CA, col="red")
1316   abline(v=CA-CA_err, col="red",lty=2)
1317   abline(v=CA+CA_err, col="red",lty=2)
1318
1319   abline(v=MA, col="blue")
1320   abline(v=MA-MA_err, col="blue",lty=2)
1321   abline(v=MA+MA_err, col="blue",lty=2)
1322
1323   print(p_KDE)
1324
1325 } else {
1326   p_KDE <- plot_KDE(plot_input,
1327                     main=paste(sample_name,": ", "De Distribution (Kernel Density Estimate)",sep=""),
1328                     summary = c("n","median","kurtosis","skewness"),
1329                     xlab = "Equivalent dose [Gy]",
1330                     output = FALSE)
1331

```

```

1332 p_KDE <- abline(v=CA, col="red")
1333 p_KDE <- abline(v=CA-CA_err, col="red",lty=2)
1334 p_KDE <- abline(v=CA+CA_err, col="red",lty=2)
1335
1336 p_KDE <- abline(v=MA, col="blue")
1337 p_KDE <- abline(v=MA-MA_err, col="blue",lty=2)
1338 p_KDE <- abline(v=MA+MA_err, col="blue",lty=2)
1339
1340 print(p_KDE)
1341
1342 }
1343
1344 if (OSL_IRSL_type[sample_iter]=="OSL"){
1345
1346   p_radial <- plot_RadialPlot(plot_input,
1347                               main=paste(sample_name," : ", "De Distribution (Radial Plot)",sep=""),
1348                               centrality = "mean.weighted",
1349                               # central.value = CA,
1350                               log.z = TRUE,
1351                               line = c(CA, MA),
1352                               line.col = c("red","blue"),
1353                               summary = c("n","mean.weighted","seabs.weighted","in.2s"),
1354                               summary.pos = "topleft",
1355                               # xlab = c("Data error (%)",
1356                               #         "Data precision"),
1357                               # ylab = "Scatter",
1358                               # zlab = "Equivalent dose [Gy]",
1359                               grid.col = "none",
1360                               pch=16)
1361
1362   print(p_radial)
1363
1364 } else {
1365
1366   p_radial <- plot_RadialPlot(plot_input,
1367                               main=paste(sample_name," : ", "De Distribution (Radial Plot)",sep=""),
1368                               centrality = "mean.weighted",
1369                               # central.value = CA,
1370                               log.z = TRUE,
1371                               # line = c(CA),
1372                               # line.col = c("red"),
1373                               line = c(CA, MA),
1374                               line.col = c("red","blue"),
1375                               summary = c("n","mean.weighted","seabs.weighted","in.2s"),
1376                               summary.pos = "topleft",
1377                               # xlab = c("Data error (%)",
1378                               #         "Data precision"),
1379                               # ylab = "Scatter",
1380                               # zlab = "Equivalent dose [Gy]",
1381                               grid.col = "none",
1382                               pch=16)
1383
1384   print(p_radial)
1385
1386 }
1387
1388 dev.off()
1389
1390
1391 #####
1392
1393 ## Build a plot of all the naturals and the growth curves all together
1394
1395 if (bin_or_xlsx[sample_iter] == 'bin'){
1396
1397   ## Decay Curves
1398   print(paste("Building decay curves for sample: ", sample_name,sep=""))
1399
1400   pos_vector = unique(bin_data@METADATA$POSITION) # all positions
1401
1402   decay_curves_index = as.numeric(rownames(subset(results$data, RC.Status == "OK"))) #accepted positions only
1403   decay_curves_data = RLum_object[decay_curves_index] # Risoe.BINfileData2RLum.Analysis(bin_data)
1404   decay_curves = data.frame(time=numeric(0),lum=numeric(0),position=character(0))
1405
1406   for (i_curve in 1:length(decay_curves_data)){
1407
1408     curves_df_temp = data.frame(time=decay_curves_data [[i_curve]]@records[[1]][,1],
1409                                  lum = decay_curves_data [[i_curve]]@records[[1]][,2],
1410                                  position = rep(paste("Position #", as.character(pos_vector[i_curve]), sep=""),
1411                                                  nrow(decay_curves_data [[i_curve]]@records[[1]][,]))
1412
1413     decay_curves = rbind(curves_df_temp, decay_curves)
1414
1415   }
1416
1417   p_decay <- ggplot(data=decay_curves,aes(x=time,y=lum, group=position)) +
1418     geom_line(aes(color=position),show.legend = FALSE) +
1419     scale_color_discrete(name = "Aliquot") +
1420     theme_bw() +
1421     labs(title = paste("A) Natural Decay Curves: ",
1422                        sample_name,
1423                        # " (" ,

```

```

1424         # mineral[sample_iter],
1425         # " ",
1426         # OSL_IRSL_type[sample_iter],
1427         # ")",
1428         sep=""),
1429     x="Measurement time (s)", y="Photons")
1430
1431 # png(file = paste(wdir,"output_plots\\", sample_name,"_decay_curves_compiled.png",sep=""),
1432 #      width = 6, height = 4, units = "in", res = 300)
1433 # print(p_decay)
1434 # dev.off()
1435
1436 ## Growth Curves
1437 print(paste("Building growth curves for sample: ", sample_name,sep=""))
1438
1439 growth_curves_index = as.numeric(rownames(subset(results$data, RC.Status == "OK")))
1440
1441 SAR_num = SAR_num_vals[sample_iter]
1442
1443 if (dim(results$LnLxTx.table[c("Dose","LxTx","LxTx.Error","Net_TnTx")])[1] != length(rep(as.character(pos_vector),each=SAR_num
1444 ))) {
1445     stop(paste("There is a shape mismatch with the bin file for ",sample_name, ".", "File name = ", file_names[sample_iter], sep="
1446     "))
1447 }
1448
1449 growth_curves_data = cbind(results$LnLxTx.table[c("Dose","LxTx","LxTx.Error","Net_TnTx")], position = rep(as.character(pos_
1450 vector),each=SAR_num), run=rep(1:SAR_num,length(pos_vector)))
1451 growth_curves_selected = rep(growth_curves_index*SAR_num, each=SAR_num) - rep(seq((SAR_num - 1), 0, -1), length(growth_curves_
1452 index))
1453 growth_curves_data_plot = growth_curves_data[growth_curves_selected,]
1454
1455 # Risoe_dose_rate_subset[rep(seq_len(nrow(Risoe_dose_rate_subset)), each=6),]
1456 #
1457 # growth_curves_data_plot[,1] = growth_curves_data_plot[,1]*Risoe_dose_rate_subset[rep(seq_len(nrow(Risoe_dose_rate_subset)),
1458 each=6),] ["IRR_DR"]
1459
1460 growth_curves_fitted = data.frame(x_fit=numeric(0),y_fit=numeric(0),position_fit=character(0))
1461
1462 iter_gc = 0
1463
1464 for (m in unique(growth_curves_data_plot$position)){
1465     iter_gc = iter_gc + 1
1466     subset_gc = subset(growth_curves_data_plot, position==m)
1467     x = seq(0,max(subset_gc$Dose),length.out=100)
1468     y_gc = eval(results$Formula[[growth_curves_index[iter_gc]]])
1469     position_gc = as.character(rep(m,length(x)))
1470
1471     temp = data.frame(x_fit=x, y_fit=y_gc, position_fit=position_gc)
1472
1473     growth_curves_fitted = rbind(growth_curves_fitted, temp)
1474 }
1475
1476 p_growth <- ggplot(data=growth_curves_data_plot, aes(x=Dose, y=LxTx, group=position, color=position)) +
1477     geom_point(show.legend = FALSE) +
1478     geom_errorbar(aes(ymin=LxTx-LxTx.Error,ymax=LxTx+LxTx.Error), width = 0.2,show.legend = FALSE) +
1479     geom_line(data=growth_curves_fitted, aes(x=x_fit,y=y_fit, group=position_fit, color=position_fit),show.
1480 legend = FALSE) +
1481     theme_bw() +
1482     labs(title = "B) Growth Curves",
1483          x="Radiation Dose (seconds)", y="Lx/Tx")
1484
1485 p_test_dose <- ggplot(data = growth_curves_data_plot, aes(x=run,y=Net_TnTx, group=position)) +
1486     geom_line() +
1487     geom_point(shape=21,fill="blue") +
1488     theme_bw() +
1489     labs(title = "C) Test Dose",
1490          x="Run Number", y="Photons")
1491
1492 ## IRSL curves
1493
1494 # data for plotting
1495
1496 # RC values
1497 Recyl_Ratio_Vals = as.numeric(unlist(subset(x = results$rejection.criteria,
1498 Criteria == "Recycling ratio (R4/R1)" & Status == "OK")["Value"])))
1499
1500 Recup_vals = as.numeric(unlist(subset(x = results$rejection.criteria,
1501 Criteria == "Recuperation rate 1" & Status == "OK")["Value"])))
1502
1503 De_errors = as.numeric(unlist(subset(x = results$rejection.criteria,
1504 Criteria == "Palaeodose error" & Status == "OK")["Value"])))
1505
1506 if ((OSL_IRSL_type[sample_iter]=="OSL")&(!is.null(RLum_object_IRSL))){
1507     print(paste("Building IRSL curves for sample: ", sample_name,sep=""))
1508 }
1509

```

```

1510 IRSL_curves_data = RLum_object_IRSL
1511 IRSL_curves = data.frame(time=numeric(0),lum=numeric(0),position=character(0))
1512
1513
1514 for (i_curve in 1:length(IRSL_curves_data)){
1515
1516   curves_IR_temp = data.frame(time=IRSL_curves_data[[i_curve]]@records[[1]][,1],
1517                                lum = IRSL_curves_data[[i_curve]]@records[[1]][,2],
1518                                position = rep(paste("Position #", as.character(pos_vector[i_curve])), sep=""),
1519                                                nrow(IRSL_curves_data[[i_curve]]@records[[1]][,]))
1520
1521   IRSL_curves = rbind(curves_IR_temp, IRSL_curves)
1522
1523 }
1524
1525 p_IRSL <- ggplot(data=IRSL_curves,aes(x=time,y=lum, group=position)) +
1526   geom_line(aes(color=position),show.legend = FALSE) +
1527   scale_color_discrete(name = "Aliquot") +
1528   theme_bw() +
1529   labs(title = "E) IRSL Detection",
1530        x="Measurement time (s)", y="Photons")
1531
1532
1533 IR_dep_vals = as.numeric(unlist(subset(x = results$rejection.criteria,
1534                                       Criteria == "Recycling ratio (R5/R1)" & Status == "OK")["Value"])) / as.numeric(unlist(
1535   subset(x = results$rejection.criteria,
1536         Criteria == "Recycling ratio (R4/R1)" & Status == "OK")["Value"]))
1537
1538 p_IR <- ggplot(data.frame(vals=IR_dep_vals),aes(x=vals)) +
1539   geom_histogram(binwidth = 0.002) +
1540   xlim(0.9,1.1) +
1541   labs(title="H) IR Dep. Ratio",x="Value", y = "Count") +
1542   theme_bw()
1543
1544 } else {
1545   p_IRSL <- ggplot() + geom_blank()
1546   p_IR <- ggplot() + geom_blank()
1547 }
1548
1549
1550 # Plots
1551
1552 p_RR <- ggplot(data.frame(vals=Recyl_Ratio_Vals),aes(x=vals)) +
1553   geom_histogram(binwidth = 0.002) +
1554   xlim(0.9,1.1) +
1555   labs(title="E) Recycling Ratio",x="Value", y = "Count") +
1556   theme_bw()
1557
1558 p_Recup <- ggplot(data.frame(vals=Recup_vals),aes(x=vals)) +
1559   geom_histogram(binwidth = 0.001) +
1560   # xlim(0.9,1.1) +
1561   labs(title="F) Recuperation",x="Value", y = "Count") +
1562   theme_bw()
1563
1564 p_De_errors <- ggplot(data.frame(vals=De_errors),aes(x=vals)) +
1565   geom_histogram(binwidth = 0.001) +
1566   # xlim(0.9,1.1) +
1567   labs(title="G) Paleodose Error",x="Value", y = "Count") +
1568   theme_bw()
1569
1570
1571
1572 png(file = paste(sample_name,"_growth_curves_compiled.png",sep=""),
1573     width = 6, height = 4, units = "in", res = 300)
1574 print(p_growth)
1575 dev.off()
1576
1577 png(file = paste(sample_name,"_growth_decay_compiled.png",sep=""),
1578     width = 8, height = 6, units = "in", res = 300)
1579 grid.arrange(p_decay,p_growth,nrow=2)
1580 dev.off()
1581
1582 png(file = paste(wdir, sample_name,"_compiled_plots.png",sep=""), #"output_plots\\",
1583     width = 8, height = 10, units = "in", res = 300)
1584 if (OSL_IRSL_type[sample_iter]=="OSL"){
1585
1586   grid.arrange(p_decay, # 1
1587                p_growth, # 2
1588                p_test_dose, # 3
1589                p_water, # 4
1590                p_RR, # 5
1591                p_Recup, # 6
1592                p_De_errors, # 7
1593                p_IR, # 8
1594                p_IRSL, # 9
1595                layout_matrix = rbind(c(1,1,2,2),
1596                                     c(3,4,4,4),
1597                                     c(5,4,4,4),
1598                                     c(6,7,8,9)))
1599 } else {
1600

```

```

1601     grid.arrange(p_decay,
1602                  p_growth,
1603                  p_test_dose,
1604                  p_water,
1605                  p_RR,
1606                  p_Recup,
1607                  p_De_errors,
1608                  p_IR,
1609                  layout_matrix = rbind(c(1,1,2,2),
1610                                         c(3,4,4,4),
1611                                         c(5,6,7,8)))
1612 }
1613 dev.off()
1614
1615 # png(file = paste(wdir, sample_name, "_compiled_plots.png", sep=""), #"output_plots\\",
1616 #       width = 8, height = 10, units = "in", res = 300)
1617 #
1618 # grid.arrange(p_decay, p_growth, p_test_dose, p_water, p_IRSL, layout_matrix = rbind(c(1,1,2,2),
1619 #                                           c(3,4,4,4),
1620 #                                           c(NA,NA,5,5)))
1621 #
1622 # dev.off()
1623
1624 #####
1625
1626 ## Build the decay curves library
1627
1628 print(paste("Building decay curves extra plots for sample: ", sample_name, sep=""))
1629
1630 pdf(file = paste(wdir, "/output_plots/", sample_name, "_decay_curves.pdf", sep=""),
1631     paper = "letter", height = 10)
1632 par(mfrow = c(2,2))
1633
1634 iter = 0
1635
1636 position_array = unlist(unique(bin_data@METADATA["POSITION"]), use.names=FALSE)
1637
1638 for (p_iter in position_array){
1639
1640     if (OSL_IRSL_type[sample_iter]=="pIRSL"){
1641
1642         temp <- Risoe.BINfileData2RLum.Analysis(bin_data, pos=p_iter, ltype = "IRSL")
1643
1644         plot_RLum.Analysis(
1645             temp,
1646             subset = list(recordType = gsub(pattern = "pIRSL", replacement = "IRSL", toString(OSL_IRSL_type[sample_iter]))),
1647             combine = TRUE,
1648             norm = FALSE,
1649             plot.single = TRUE,
1650             legend = TRUE,
1651             legend.text = paste(as.character(as.list((bin_data@METADATA["IRR_TIME"][(iter*12 + (1:12)),1])), "s B", sep=""),
1652                                 main = paste(sample_name, "position number =", toString(position_array[p_iter]), sep=" ")
1653         )
1654     }
1655 }
1656
1657 if (OSL_IRSL_type[sample_iter]=="IRSL"){
1658
1659     temp <- Risoe.BINfileData2RLum.Analysis(bin_data, pos=p_iter, ltype = "IRSL")
1660
1661     plot_RLum.Analysis(
1662         temp,
1663         subset = list(recordType = gsub(pattern = "IRSL", replacement = "IRSL", toString(OSL_IRSL_type[sample_iter]))),
1664         combine = TRUE,
1665         norm = FALSE,
1666         plot.single = TRUE,
1667         legend = TRUE,
1668         legend.text = paste(as.character(as.list((bin_data@METADATA["IRR_TIME"][(iter*12 + (1:12)),1])), "s B", sep=""),
1669                             main = paste(sample_name, "position number =", toString(position_array[p_iter]), sep=" ")
1670     )
1671 }
1672
1673 if (OSL_IRSL_type[sample_iter]=="OSL"){
1674
1675     temp <- Risoe.BINfileData2RLum.Analysis(bin_data, pos=p_iter, ltype = "OSL")
1676
1677     plot_RLum.Analysis(
1678         temp,
1679         subset = list(recordType = gsub(pattern = "OSL", replacement = "OSL", toString(OSL_IRSL_type[sample_iter]))),
1680         combine = TRUE,
1681         norm = FALSE,
1682         plot.single = TRUE,
1683         legend = TRUE,
1684         legend.text = paste(as.character(as.list((bin_data@METADATA["IRR_TIME"][(iter*12 + (1:12)),1])), "s B", sep=""),
1685                             main = paste(sample_name, "position number =", toString(position_array[p_iter]), sep=" ")
1686     )
1687 }
1688
1689 iter = iter+1
1690
1691
1692

```

```

1693 }
1694
1695 dev.off()
1696
1697 #####
1700 ## Build the growth curves library
1701
1702 print(paste("Building growth curves extra plots for sample: ", sample_name, sep=""))
1703
1704 pdf(file = paste(wdir, "/output_plots/", sample_name, "_growth_curves.pdf", sep=""),
1705     paper = "letter", height = 10)
1706 par(mfrow = c(3,2))
1707
1708 for (k in 0:((nrow(results["LnLxTnTx.table"])/6)-1)){
1709
1710     index = k*6 + 1:6
1711
1712     x1 = results["LnLxTnTx.table"]["Dose"][index,1]
1713     x2 = results["LnLxTnTx.table"]["LxTx"][index,1]
1714     x3 = results["LnLxTnTx.table"]["LxTx.Error"][index,1]
1715     x4 = results["LnLxTnTx.table"]["TnTx"][index,1]
1716
1717     plot_GrowthCurve(data.frame(x=x1,y=x2,z=x3), na.rm = TRUE, mode = "interpolation",
1718                      fit.method = "EXP", fit.force_through_origin = TRUE,
1719                      fit.weights = FALSE, fit.includingRepeatedRegPoints = TRUE,
1720                      fit.NumberRegPoints = NULL, fit.NumberRegPointsReal = NULL,
1721                      fit.bounds = TRUE, NumberIterations.MC = 100, legend=FALSE, output.plot = TRUE,
1722                      xlab="Dose [Gy]",
1723                      output.plotExtended = FALSE, output.plotExtended.single = TRUE,
1724                      cex.global = 1, txtProgressBar = TRUE, verbose = FALSE,
1725                      main = paste("Sample:", sample_name, ", Aliquot", toString(k+1), sep=" "))
1726
1727     plot((1:6), x4/x4[1],
1728          type="b",
1729          main="Test Dose Response",
1730          xlab="Run Number",
1731          ylab="Normalized Test Dose",
1732          ylim= c((1 - 2*max(abs(1 - x4/x4[1]))), (1 + 2*max(abs(1 - x4/x4[1])))))
1733
1734     abline(h=1,col="black")
1735
1736 }
1737
1738 dev.off()
1739
1740 } else {
1741
1742     png(file = paste(wdir, sample_name, "_compiled_plots.png", sep=""), #"output_plots\\",
1743         width = 8, height = 6, units = "in", res = 300)
1744
1745     print(p_water)
1746
1747     dev.off()
1748
1749 }
1750
1751 }}
1752 #####
1753
1754 ## Proper formatting
1755
1756 print("Conducting correct formatting")
1757
1758 ## K values
1759 pretty_K_below_10 = paste(sprintf("%#.3g", K_values[K_values<10]), " $\\pm$ ",
1760                            formatC(K_err[K_values<10], digits = 2, format = "f"),
1761                            sep="")
1762 pretty_K_above_10 = paste(sprintf("%#.3g", K_values[K_values>=10]), " $\\pm$ ",
1763                            formatC(K_err[K_values>=10], digits = 2),
1764                            sep="")
1765
1766 pretty_K_below_1 = paste(sprintf("%#.2g", K_values[K_values<1]), " $\\pm$ ",
1767                           formatC(K_err[K_values<1], digits = 2),
1768                           sep="")
1769
1770 pretty_K = as.character(K_values) # dummy values
1771 pretty_K[K_values<10] = pretty_K_below_10
1772 pretty_K[K_values>=10] = pretty_K_above_10
1773 pretty_K[K_values<1] = pretty_K_below_1
1774
1775 ## U values
1776 pretty_U_below_10 = paste(formatC(U_values[U_values<10], digits = 2, format = "f"), " $\\pm$ ",
1777                            formatC(U_err[U_values<10], digits = 2, format = "f"),
1778                            sep="")
1779 pretty_U_above_10 = paste(formatC(U_values[U_values>=10], digits = 1, format = "f"), " $\\pm$ ",
1780                            formatC(U_err[U_values>=10], digits = 1, format = "f"),
1781                            sep="")
1782
1783 pretty_U_below_1 = paste(sprintf("%#.2g", U_values[U_values<1]), " $\\pm$ ",
1784

```



```

1785         formatC(U_err[U_values<1], digits = 2),
1786         sep="")
1787
1788 pretty_U = as.character(U_values) # dummy values
1789 pretty_U[U_values<10] = pretty_U_below_10
1790 pretty_U[U_values>=10] = pretty_U_above_10
1791 pretty_U[U_values<1] = pretty_U_below_1
1792
1793
1794 ## Th Values
1795 pretty_Th_below_10 = paste(formatC(Th_values[Th_values<10], digits = 2, format = "f"), " $\\pm$ ",
1796                             formatC(Th_err[Th_values<10], digits = 2, format = "f"),
1797                             sep="")
1798 pretty_Th_above_10 = paste(formatC(Th_values[Th_values>=10], digits = 1, format = "f"), " $\\pm$ ",
1799                             formatC(Th_err[Th_values>10], digits = 1, format = "f"),
1800                             sep="")
1801
1802 pretty_Th_below_1 = paste(sprintf("%.2g", Th_values[Th_values<1]), " $\\pm$ ",
1803                             formatC(Th_err[Th_values<1], digits = 2),
1804                             sep="")
1805
1806 pretty_Th = as.character(Th_values) # dummy values
1807 pretty_Th[Th_values<10] = pretty_Th_below_10
1808 pretty_Th[Th_values>=10] = pretty_Th_above_10
1809 pretty_Th[Th_values<1] = pretty_U_below_1
1810
1811 ## DR Values
1812
1813 DR_values = dose_rates
1814 DR_err = dose_rates_err
1815
1816 pretty_DR_below_1 = paste(formatC(DR_values[DR_values<1], digits = 2, format = "f"), " $\\pm$ ",
1817                             formatC(DR_err[DR_values<1], digits = 2, format = "f"),
1818                             sep="")
1819 pretty_DR_above_1 = paste(formatC(DR_values[DR_values>=1], digits = 2, format = "f"), " $\\pm$ ",
1820                             formatC(DR_err[DR_values>1], digits = 2, format = "f"),
1821                             sep="")
1822 pretty_DR = as.character(DR_values) # dummy values
1823 pretty_DR[DR_values<1] = pretty_DR_below_1
1824 pretty_DR[DR_values>=1] = pretty_DR_above_1
1825
1826 ## CAM De values
1827
1828 CAM_DE_values = final_xlsx_table["de_cam"][,]
1829 CAM_DE_err = final_xlsx_table["de_cam_err"][,]
1830
1831 CAM_DE_values[is.na(CAM_DE_values)] = 0.0
1832 CAM_DE_err[is.na(CAM_DE_err)] = 0.0
1833
1834 pretty_CAM_DE_below_100 = paste(formatC(CAM_DE_values[CAM_DE_values<100], digits = 1, format = "f"), " $\\pm$ ",
1835                                     formatC(CAM_DE_err[CAM_DE_values<100], digits = 1, format = "f"),
1836                                     sep="")
1837 pretty_CAM_DE_above_100 = paste(formatC(CAM_DE_values[CAM_DE_values>=100], digits = 0, format = "f"), " $\\pm$ ",
1838                                     formatC(CAM_DE_err[CAM_DE_values>=100], digits = 0, format = "f"),
1839                                     sep="")
1840
1841 pretty_CAM_DE_below_10 = paste(formatC(CAM_DE_values[CAM_DE_values<10], digits = 2, format = "f"), " $\\pm$ ",
1842                                     formatC(CAM_DE_err[CAM_DE_values<10], digits = 2, format = "f"),
1843                                     sep="")
1844
1845 pretty_CAM_DE = as.character(CAM_DE_values) # dummy values
1846 pretty_CAM_DE[CAM_DE_values<100] = pretty_CAM_DE_below_100
1847 pretty_CAM_DE[CAM_DE_values>=100] = pretty_CAM_DE_above_100
1848 pretty_CAM_DE[CAM_DE_values<10] = pretty_CAM_DE_below_10
1849
1850 ## MAM De values
1851
1852 MAM_DE_values = final_xlsx_table["de_mam"][,]
1853 MAM_DE_err = final_xlsx_table["de_mam_err"][,]
1854
1855 MAM_DE_values[is.na(MAM_DE_values)] = 0.0
1856 MAM_DE_err[is.na(MAM_DE_err)] = 0.0
1857
1858 pretty_MAM_DE_below_100 = paste(formatC(MAM_DE_values[MAM_DE_values<100], digits = 1, format = "f"), " $\\pm$ ",
1859                                     formatC(MAM_DE_err[MAM_DE_values<100], digits = 1, format = "f"),
1860                                     sep="")
1861 pretty_MAM_DE_above_100 = paste(formatC(MAM_DE_values[MAM_DE_values>=100], digits = 0, format = "f"), " $\\pm$ ",
1862                                     formatC(MAM_DE_err[MAM_DE_values>=100], digits = 0, format = "f"),
1863                                     sep="")
1864
1865 pretty_MAM_DE_below_10 = paste(formatC(MAM_DE_values[MAM_DE_values<10], digits = 2, format = "f"), " $\\pm$ ",
1866                                     formatC(MAM_DE_err[MAM_DE_values<10], digits = 2, format = "f"),
1867                                     sep="")
1868
1869 pretty_MAM_DE = as.character(MAM_DE_values) # dummy values
1870 pretty_MAM_DE[MAM_DE_values<100] = pretty_MAM_DE_below_100
1871 pretty_MAM_DE[MAM_DE_values>=100] = pretty_MAM_DE_above_100
1872 pretty_MAM_DE[MAM_DE_values<10] = pretty_MAM_DE_below_10
1873
1874
1875 ## CAM Age values
1876

```

```

1877 CAM_age_values = final_xlsx_table["age_cam"][,]
1878 CAM_age_err = final_xlsx_table["age_cam_err"][,]
1879
1880 CAM_age_values[is.na(CAM_age_values)] = 0.0
1881 CAM_age_err[is.na(CAM_age_err)] = 0.0
1882
1883 pretty_CAM_age_below_100 = paste(formatC(CAM_age_values[CAM_age_values<100], digits = 1, format = "f")," $\\pm$ ",
1884                                   formatC(CAM_age_err[CAM_age_values<100], digits = 1, format = "f"),
1885                                   sep="")
1886 pretty_CAM_age_above_100 = paste(formatC(CAM_age_values[CAM_age_values>=100], digits = 0, format = "f")," $\\pm$ ",
1887                                   formatC(CAM_age_err[CAM_age_values>=100], digits = 0, format = "f"),
1888                                   sep="")
1889
1890 pretty_CAM_age_below_10 = paste(formatC(CAM_age_values[CAM_age_values<10], digits = 2, format = "f")," $\\pm$ ",
1891                                   formatC(CAM_age_err[CAM_age_values<10], digits = 2, format = "f"),
1892                                   sep="")
1893
1894 pretty_CAM_age = as.character(CAM_age_values) # dummy values
1895 pretty_CAM_age[CAM_age_values<100] = pretty_CAM_age_below_100
1896 pretty_CAM_age[CAM_age_values>=100] = pretty_CAM_age_above_100
1897 pretty_CAM_age[CAM_age_values<10] = pretty_CAM_age_below_10
1898
1899 ## CAM Age fade corrected values
1900
1901 CAM_age_fade_values = final_xlsx_table["age_cam_fade"][,]
1902 CAM_age_fade_err = final_xlsx_table["age_cam_fade_err"][,]
1903
1904 CAM_age_fade_values[is.na(CAM_age_fade_values)] = 0.0
1905 CAM_age_fade_err[is.na(CAM_age_fade_err)] = 0.0
1906
1907 pretty_CAM_age_fade_below_100 = paste(formatC(CAM_age_fade_values[CAM_age_fade_values<100], digits = 1, format = "f")," $\\pm$ ",
1908                                   formatC(CAM_age_fade_err[CAM_age_fade_values<100], digits = 1, format = "f"),
1909                                   sep="")
1910 pretty_CAM_age_fade_above_100 = paste(formatC(CAM_age_fade_values[CAM_age_fade_values>=100], digits = 0, format = "f")," $\\pm$ ",
1911                                   formatC(CAM_age_fade_err[CAM_age_fade_values>=100], digits = 0, format = "f"),
1912                                   sep="")
1913
1914 pretty_CAM_age_fade_below_10 = paste(formatC(CAM_age_fade_values[CAM_age_fade_values<10], digits = 2, format = "f")," $\\pm$ ",
1915                                   formatC(CAM_age_fade_err[CAM_age_fade_values<10], digits = 2, format = "f"),
1916                                   sep="")
1917
1918 pretty_CAM_age_fade = as.character(CAM_age_fade_values) # dummy values
1919 pretty_CAM_age_fade[CAM_age_fade_values<100] = pretty_CAM_age_fade_below_100
1920 pretty_CAM_age_fade[CAM_age_fade_values>=100] = pretty_CAM_age_fade_above_100
1921 pretty_CAM_age_fade[CAM_age_fade_values<10] = pretty_CAM_age_fade_below_10
1922
1923
1924 ## MAM Age values
1925
1926 MAM_age_values = final_xlsx_table["age_mam"][,]
1927 MAM_age_err = final_xlsx_table["age_mam_err"][,]
1928
1929 MAM_age_values[is.na(MAM_age_values)] = 0.0
1930 MAM_age_err[is.na(MAM_age_err)] = 0.0
1931
1932 pretty_MAM_age_below_100 = paste(formatC(MAM_age_values[MAM_age_values<100], digits = 1, format = "f")," $\\pm$ ",
1933                                   formatC(MAM_age_err[MAM_age_values<100], digits = 1, format = "f"),
1934                                   sep="")
1935 pretty_MAM_age_above_100 = paste(formatC(MAM_age_values[MAM_age_values>=100], digits = 0, format = "f")," $\\pm$ ",
1936                                   formatC(MAM_age_err[MAM_age_values>=100], digits = 0, format = "f"),
1937                                   sep="")
1938
1939 pretty_MAM_age_below_10 = paste(formatC(MAM_age_values[MAM_age_values<10], digits = 2, format = "f")," $\\pm$ ",
1940                                   formatC(MAM_age_err[MAM_age_values<10], digits = 2, format = "f"),
1941                                   sep="")
1942
1943 pretty_MAM_age = as.character(MAM_age_values) # dummy values
1944 pretty_MAM_age[MAM_age_values<100] = pretty_MAM_age_below_100
1945 pretty_MAM_age[MAM_age_values>=100] = pretty_MAM_age_above_100
1946 pretty_MAM_age[MAM_age_values<10] = pretty_MAM_age_below_10
1947
1948
1949 ## MAM Age fade corrected values
1950
1951 MAM_age_fade_values = final_xlsx_table["age_mam_fade"][,]
1952 MAM_age_fade_err = final_xlsx_table["age_mam_fade_err"][,]
1953
1954 MAM_age_fade_values[is.na(MAM_age_fade_values)] = 0.0
1955 MAM_age_fade_err[is.na(MAM_age_fade_err)] = 0.0
1956
1957 pretty_MAM_age_fade_below_100 = paste(formatC(MAM_age_fade_values[MAM_age_fade_values<100], digits = 1, format = "f")," $\\pm$ ",
1958                                   formatC(MAM_age_fade_err[MAM_age_fade_values<100], digits = 1, format = "f"),
1959                                   sep="")
1960 pretty_MAM_age_fade_above_100 = paste(formatC(MAM_age_fade_values[MAM_age_fade_values>=100], digits = 0, format = "f")," $\\pm$ ",
1961                                   formatC(MAM_age_fade_err[MAM_age_fade_values>=100], digits = 0, format = "f"),
1962                                   sep="")
1963
1964 pretty_MAM_age_fade_below_10 = paste(formatC(MAM_age_fade_values[MAM_age_fade_values<10], digits = 2, format = "f")," $\\pm$ ",
1965                                   formatC(MAM_age_fade_err[MAM_age_fade_values<10], digits = 2, format = "f"),
1966                                   sep="")
1967
1968 pretty_MAM_age_fade = as.character(MAM_age_fade_values) # dummy values

```

```

1969 pretty_MAM_age_fade[MAM_age_fade_values<100] = pretty_MAM_age_fade_below_100
1970 pretty_MAM_age_fade[MAM_age_fade_values>=100] = pretty_MAM_age_fade_above_100
1971 pretty_MAM_age_fade[MAM_age_fade_values<10] = pretty_MAM_age_fade_below_10
1972
1973
1974 ## put into table
1975
1976 final_pretty_table = data.frame(sample_name=sample_list,
1977                                latitude = latitude_values,
1978                                longitude = longitude_values,
1979                                elevation = round(elevation_values, 0),
1980                                depths= round(depth_of_samples,1),
1981                                water_content = paste(as.character(round(field_water_content_of_samples)),
1982                                                       " (", as.character(round(saturation_water_content_of_samples)),")",
1983                                                       " [" , as.character(round(preferred_water_content_of_samples)),"]",sep=""),
1984                                k_percent = pretty_K,
1985                                uranium_ppm = pretty_U,
1986                                thorium_ppm = pretty_Th,
1987                                dose_rate = pretty_DR,
1988                                num_aliquots = paste(as.character(num_accepted),
1989                                                       " (",
1990                                                       as.character(num_total),
1991                                                       ")",
1992                                                       sep=""),
1993                                scatter = final_xlsx_table["scatter"],
1994                                de_cam = pretty_CAM_DE,
1995                                de_mam = pretty_MAM_DE,
1996                                age_cam = pretty_CAM_age,
1997                                age_mam = pretty_MAM_age,
1998                                age_cam_fade = pretty_CAM_age_fade,
1999                                age_mam_fade = pretty_MAM_age_fade)
2000
2001 # final_misc_table_3[c("age_cam", "age_cam_fade", "age_mam", "age_mam_fade")][sample_iter,] =
2002 #   c(paste(sprintf("%.3g",CA_age)," $\\pm$ ",formatC(signif(CA_age_uncert,3), format = "f", digits= 1,drop0trailing = FALSE, flag
2003 #     = "#"), sep=""),
2004 #     paste(sprintf("%.3g",CA_age_fade)," $\\pm$ ",formatC(signif(CA_age_fade_uncert,3), format = "f", digits= 1,drop0trailing =
2005 #     FALSE, flag="#"), sep=""),
2006 #     paste(sprintf("%.3g",MA_age)," $\\pm$ ",formatC(signif(MA_age_uncert,3), format = "f", digits= 1,drop0trailing = FALSE, flag
2007 #     = "#"), sep=""),
2008 #     paste(sprintf("%.3g",MA_age_fade)," $\\pm$ ",formatC(signif(MA_age_fade_uncert,3), format = "f", digits= 1,drop0trailing =
2009 #     FALSE, flag="#"), sep=""))
2010
2011 #####
2012
2013 ## Generate IPDS Data Release files
2014
2015 IPDS_data_categories = c(
2016   "sample_name",
2017   "latitude",
2018   "longitude",
2019   "elevation",
2020   "sampling_depth",
2021   "field_water_content",
2022   "saturated_water_content",
2023   "preferred_water_content_of_samples",
2024   "potassium_percent",
2025   "potassium_percent_err",
2026   "uranium_ppm",
2027   "uranium_ppm_err",
2028   "thorium_ppm",
2029   "thorium_ppm_err",
2030   "cosmic_dose_rate",
2031   "cosmic_dose_rate_err",
2032   "dose_rate",
2033   "dose_rate_err",
2034   "analysis_type",
2035   "grain_size_min",
2036   "grain_size_max",
2037   "num_aliquots_passed",
2038   "num_aliquots_total",
2039   "scatter",
2040   "de_cam",
2041   "de_cam_err",
2042   "de_mam",
2043   "de_mam_err",
2044   "age_cam",
2045   "age_cam_err",
2046   "age_mam",
2047   "age_mam_err",
2048   "age_cam_fade",
2049   "age_cam_fade_err",
2050   "age_mam_fade",
2051   "age_mam_fade_err"
2052 )
2053
2054 IPDS_data_units = c(
2055   "n/a",
2056   "Degrees",
2057   "Degrees",
2058   "Meters",
2059   "Meters",
2060   "Percentage",

```

```

2057 "Percentage",
2058 "Percentage",
2059 "Percentage",
2060 "Percentage",
2061 "Parts per Million",
2062 "Parts per Million",
2063 "Parts per Million",
2064 "Parts per Million",
2065 "Gray per Ka",
2066 "Gray per Ka",
2067 "Gray per Ka",
2068 "Gray per Ka",
2069 "n/a",
2070 "Microns",
2071 "Microns",
2072 "n/a",
2073 "n/a",
2074 "Percentage",
2075 "Gray",
2076 "Gray",
2077 "Gray",
2078 "Gray",
2079 "Kiloannum",
2080 "Kiloannum",
2081 "Kiloannum",
2082 "Kiloannum",
2083 "Kiloannum",
2084 "Kiloannum",
2085 "Kiloannum",
2086 "Kiloannum")
2087
2088 IPDS_data_descriptions <- c(
2089 "The laboratory code",
2090 "The latitude of the sample location in decimal degrees.",
2091 "The longitude of the sample location in decimal degrees.",
2092 "The elevation of the sample location in meters above sea level.",
2093 "The sampling depth below the depositional surface.",
2094 "The water content in weight percent of the sample as initially extracted from the sample container.",
2095 "The water content in weight percent of the sample after the sample has been saturated with water.",
2096 "The water content used in the actual age calculation.",
2097 "The weight percent of potassium as measured by ICP-MS or Gamma spectrometry.",
2098 "The uncertainty (in weight percent) of the potassium weight percent.",
2099 "The concentration of uranium in parts per million (ppm) as measured by ICP-MS or Gamma spectrometry.",
2100 "The uncertainty (in ppm) of the uranium concentration.",
2101 "The concentration of thorium in parts per million (ppm) as measured by ICP-MS or Gamma spectrometry.",
2102 "The uncertainty (in ppm) of the thorium concentration.",
2103 "The contribution of cosmic radiation to the background ionizing dose rate calculated by DRAC (in Gray/ka).",
2104 "The uncertainty in the cosmic dose rate (in Gray/ka).",
2105 "The background ionizing dose rate as calculated by DRAC (in Gray/ka).",
2106 "The uncertainty in the dose rate (in Gray/ka).",
2107 "The type of luminescence used for the sample. ",
2108 "The minimum grain size of the treated sample in microns.",
2109 "The maximum grain size of the treated sample in microns.",
2110 "The number of aliquots measured that passed acceptance criteria.",
2111 "The total number of aliquots measured.",
2112 "The 'overdispersion' as calculated from application of the central age model.",
2113 "The equivalent dose determined using the central age model (in Grays).",
2114 "The uncertainty in the central age model equivalent dose at 95% confidence interval (in Grays)",
2115 "The equivalent dose determined using the minimum age model (in Grays).",
2116 "The uncertainty in the minimum age model equivalent dose at 95% confidence interval (in Grays)",
2117 "The luminescence age of the sample using the central age model equivalent dose (in ka).",
2118 "The uncertainty of the central age model age at 95% confidence interval (in ka).",
2119 "The luminescence age of the sample using the minimum age model equivalent dose (in ka).",
2120 "The uncertainty of the minimum age model age at 95% confidence interval (in ka).",
2121 "The fading-corrected luminescence age of the sample using the central age model equivalent dose (in ka).",
2122 "The fading-corrected uncertainty of the central age model age at 95% confidence interval (in ka).",
2123 "The fading-corrected luminescence age of the sample using the minimum age model equivalent dose (in ka).",
2124 "The fading-corrected uncertainty of the minimum age model age at 95% confidence interval (in ka).")
2125 )
2126
2127
2128 IPDS_data_dict = data.frame(Table_Name = rep('luminescence_data.csv',length(IPDS_data_catagories)),
2129                             Field_Name = IPDS_data_catagories,
2130                             Description = IPDS_data_descriptions,
2131                             Units = IPDS_data_units,
2132                             Valid_Values_or_Range =paste(sapply(final_xlsx_table,min,na.rm = TRUE),"-",sapply(final_xlsx_table,max
,na.rm = TRUE)))
2133
2134 IPDS_data_dict[which(IPDS_data_dict$Field_Name=="sample_name"),"Valid_Values_or_Range"] <- "n/a"
2135 IPDS_data_dict[which(IPDS_data_dict$Field_Name=="scatter"),"Valid_Values_or_Range"] <- paste(min(as.numeric(gsub("%","",final_xlsx
_table$scatter))),
2136                                                "-",
2137                                                max(as.numeric(gsub("%","",final_xlsx
_table$scatter))))
2138
2139 IPDS_data <- final_xlsx_table
2140 IPDS_data$scatter <- gsub(pattern="%",replacement = "",x=IPDS_data$scatter)
2141
2142
2143 write.csv(IPDS_data, file = 'luminescence_data.csv',row.names = FALSE)
2144 write.table(IPDS_data_dict, file = 'luminescence_data_dict.csv',
2145             row.names = FALSE, col.names = c("Table Name","Field Name","Description","Units","Valid Values or Range")

```

```

2146         ), append = FALSE, na='Null', sep=',')
2147
2148 file.move(files = c('luminescence_data.csv','luminescence_data_dict.csv'),
2149 destinations = paste(wdir,"To Submit\\Data Release\\",sep=""), overwrite = TRUE)
2150
2151 #####
2152
2153 ## Output Xlsx file
2154
2155 write.xlsx(final_xlsx_table, "results_table.xlsx", sheetName="Sheet1")
2156 write.xlsx(final_summary_table, "summary_table.xlsx", sheetName="Summary Table")
2157
2158 ## output Tex table on Dose Rates
2159
2160 if (file.exists('DR_table.tex')) {
2161   file.remove('DR_table.tex')}
2162
2163 sink('DR_table.tex', append=FALSE)
2164 print(xtable(final_pretty_table[c("sample_name",
2165   "elevation",
2166   "depths",
2167   "water_content",
2168   "k_percent",
2169   "uranium_ppm",
2170   "thorium_ppm",
2171   "dose_rate")],
2172   auto=TRUE), include.rownames=FALSE)
2173 sink()
2174
2175 current_file <- readLines("DR_table.tex")
2176 current_file <- gsub(pattern = "\\$\\backslash$pm\\$", replace = "$ \\pm $", x = current_file,fixed=TRUE)
2177 current_file <- head(current_file,-3)
2178 current_file <- current_file[-(1:8)]
2179
2180 writeLines(current_file,"DR_table.tex")
2181
2182 ## output tex table on Equivalent Dose
2183
2184 if (file.exists('De_table.tex')) {
2185   file.remove('De_table.tex')}
2186
2187 sink('De_table.tex', append=FALSE)
2188 print(xtable(final_pretty_table[c("sample_name",
2189   "num_aliquots",
2190   "scatter",
2191   "de_cam",
2192   "de_mam",
2193   "age_cam",
2194   "age_mam")],
2195   auto=TRUE),include.rownames=FALSE)
2196 sink()
2197
2198 current_file <- readLines("De_table.tex")
2199 current_file <- gsub(pattern = "\\$\\backslash$pm\\$", replace = "$ \\pm $", x = current_file,fixed=TRUE)
2200 current_file <- head(current_file,-3)
2201 current_file <- current_file[-(1:8)]
2202
2203 writeLines(current_file,"De_table.tex")
2204
2205 ## output tex table on Misc data 1
2206
2207 if (file.exists('misc_table_1.tex')) {
2208   file.remove('misc_table_1.tex')}
2209
2210 sink('misc_table_1.tex', append=FALSE)
2211 print(xtable(final_misc_table_1, auto=TRUE),include.rownames=FALSE)
2212 sink()
2213
2214 current_file <- readLines('misc_table_1.tex')
2215 current_file <- head(current_file,-3)
2216 current_file <- current_file[-(1:8)]
2217
2218 writeLines(current_file,'misc_table_1.tex')
2219
2220 ## output tex table on Misc data 2
2221
2222 if (file.exists('misc_table_2.tex')) {
2223   file.remove('misc_table_2.tex')}
2224
2225 sink('misc_table_2.tex', append=FALSE)
2226 print(xtable(final_misc_table_2, auto=TRUE),include.rownames=FALSE)
2227 sink()
2228
2229 current_file <- readLines('misc_table_2.tex')
2230 current_file <- gsub(pattern = "\\$\\backslash$pm\\$", replace = "$ \\pm $", x = current_file,fixed=TRUE)
2231 current_file <- head(current_file,-3)
2232 current_file <- current_file[-(1:8)]
2233
2234 writeLines(current_file,'misc_table_2.tex')
2235

```

```

2238
2239 # output tex table on misc data 3
2240
2241 if (file.exists('misc_table_3.tex')) {
2242   file.remove('misc_table_3.tex')}
2243
2244 sink('misc_table_3.tex', append=FALSE)
2245 print(xtable(final_misc_table_3, auto=TRUE),include.rownames=FALSE)
2246 sink()
2247
2248 current_file <- readLines('misc_table_3.tex')
2249 current_file <- gsub(pattern = "\\$\\backslash$pm\\$", replace = "$ \\pm $", x = current_file,fixed=TRUE)
2250 current_file <- head(current_file,-3)
2251 current_file <- current_file[-(1:8)]
2252
2253 writeLines(current_file,'misc_table_3.tex')
2254
2255 # output tex table on misc data 3
2256
2257 if (file.exists('De_table_feldspar.tex')) {
2258   file.remove('De_table_feldspar.tex')}
2259
2260 de_table_feldspar = cbind(final_pretty_table[c("sample_name",
2261                                               "num_aliquots",
2262                                               "scatter",
2263                                               "de_cam",
2264                                               "age_cam",
2265                                               "age_cam_fade")]))
2266
2267 sink('De_table_feldspar.tex', append=FALSE)
2268 print(xtable(de_table_feldspar,auto=TRUE),include.rownames=FALSE)
2269 sink()
2270
2271 current_file <- readLines('De_table_feldspar.tex')
2272 current_file <- gsub(pattern = "\\$\\backslash$pm\\$", replace = "$ \\pm $", x = current_file,fixed=TRUE)
2273 current_file <- head(current_file,-3)
2274 current_file <- current_file[-(1:8)]
2275
2276 writeLines(current_file,'De_table_feldspar.tex')
2277
2278 if (file.exists('De_table_feldspar_MAM.tex')) {
2279   file.remove('De_table_feldspar_MAM.tex')}
2280
2281 de_table_feldspar = cbind(final_pretty_table[c("sample_name",
2282                                               "num_aliquots",
2283                                               "scatter",
2284                                               "de_mam",
2285                                               "age_mam",
2286                                               "age_mam_fade")]))
2287
2288 sink('De_table_feldspar_MAM.tex', append=FALSE)
2289 print(xtable(de_table_feldspar,auto=TRUE),include.rownames=FALSE)
2290 sink()
2291
2292 current_file <- readLines('De_table_feldspar_MAM.tex')
2293 current_file <- gsub(pattern = "\\$\\backslash$pm\\$", replace = "$ \\pm $", x = current_file,fixed=TRUE)
2294 current_file <- head(current_file,-3)
2295 current_file <- current_file[-(1:8)]
2296
2297 writeLines(current_file,'De_table_feldspar_MAM.tex')
2298
2299 ## Generate text for a tex file to describe the samples
2300
2301 if (file.exists("samples.tex")) {
2302   file.remove("samples.tex")}
2303
2304 TeXinput = file("samples.tex")
2305 #sink("samples.tex", append = TRUE)
2306
2307 sample_iter = 0
2308
2309 for (sample_name in sample_list){
2310
2311   sample_iter = sample_iter +1
2312
2313   line1=as.character(paste("\\subsection{" ,sample_name," Results}",sep=""))
2314   line2=as.character(paste("\\begin{figure}[h] ", sep=""))
2315   line3=as.character(paste("\\centering"))
2316   line4=as.character(paste("\\addtolength{\\leftskip}{-2cm}"))
2317   line5=as.character(paste("\\addtolength{\\rightskip}{-2cm}"))
2318   line6=as.character(paste("\\includegraphics[scale=0.5]{", sample_name,"_plots.png",sep=""))
2319   line7=as.character(paste("\\caption[" ,sample_name," $D_E$ KDE and Radial Plot]{Plots of Equivalent Dose for ",sample_name,". Red
2320     lines shows CAM equivalent dose, blue lines show MAM equivalent dose, solid line shows the age, and dashed lines show
2321     uncertainty. \\label{fig:",sample_name,"} ", sep=""))
2322   line8=as.character(paste("\\end{figure}", sep=""))
2323
2324   write(c(line1,line2,line3,line4,line5,line6,line7,line8),file="samples.tex",append=TRUE)
2325
2326
2327

```

```

2328 line2=as.character(paste("\\begin{figure}[h]", sep=""))
2329 line3=as.character(paste("\\centering"))
2330 line4=as.character(paste("\\addtolength{\\leftskip}{-2cm}"))
2331 line5=as.character(paste("\\addtolength{\\rightskip}{-2cm}"))
2332 line6=as.character(paste("\\includegraphics[width=\\textwidth]{", sample_name, "_compiled_plots.png", sep=""))
2333 # scale=0.175
2334
2335 line7=as.character(paste("\\caption[" ,sample_name, " luminescence characteristics]{Plots of luminescence characteristics for ",
sample_name,
2336 " . A) Compilation of the natural luminescence decay curves for each aliquot. ",
2337 "B) Compilation of the growth curves for each aliquot generated by the SAR protocol (Table \\ref{tab:RC
}) . ",
2338 "C) Magnitude of test dose response for each SAR cycle. ",
2339 "D) Graph of CAM or MAM age as a function of assumed water content. Vertical black lines show the Field
water content, the saturated water content, and our chosen (modeled) water content. ",
2340 "E,F,G) Histograms of rejection criteria values. ",
2341 "H) Histogram of IRSL Depletion Ratios. ",
2342 "I) IRSL observed during depletion ratio test. Note H, and I not present for all samples. ",
2343 "\\label{fig:", sample_name, "_compiled}}", sep=""))
2344 line8=as.character(paste("\\end{figure}", sep=""))
2345
2346 write(c(line2,line3,line4,line5,line6,line7,line8),file="samples.tex",append=TRUE)
2347
2348
2349 input_text = paste("We analyzed ",toString(sample_name), " using ", toString(analysis_type[sample_iter])," on ",
2350 toString(grain_size_min[sample_iter]),"-",toString(grain_size_max[sample_iter]),
2351 " $\\mu$ m$ size grains. Out of ",toString(num_total[sample_iter])," aliquots, ",
2352 toString(num_accepted[sample_iter]),
2353 " produced acceptable luminescence characteristics and passed acceptance criteria as listed in Table \\ref{
tab:RC}.",
2354 " Aliquots that passed acceptance criteria are plotted as a kernel density estimate and as a radial plot in
Figure \\ref{fig:",toString(sample_name),"}.",
2355 " Application of the central age model (CAM, Galbraith et al., 1999) produced an equivalent dose of ",
2356 toString(final_results_table["de_cam"][sample_iter,1]), " Gray, ",
2357 "which leads to a CAM age of ", toString(final_results_table["age_cam"][sample_iter,1])," ka.",
2358 " In contrast, application of the minimum age model (MAM, Galbraith et al., 1999) produces an equivalent dose
of ",
2359 toString(final_results_table["de_mam"][sample_iter,1]), " Gray, ",
2360 "which leads to a MAM age of ", toString(final_results_table["age_mam"][sample_iter,1]), " ka.",
2361 sep="")
2362 input_text = paste(input_text,"\\clearpage", sep=" ")
2363 # input_text = paste(input_text, "\\newline", "\\clearpage", notes[sample_iter], sep=" ")
2364
2365 write(input_text,file="samples.tex",append=TRUE)
2366
2367 }
2368 close(TeXinput)
2369
2370 graphics.off()
2371
2372 # end of code
2373
2374 #
2375 # system("xelatex.exe -synctex=1 -interaction=nonstopmode report.tex")
2376 # system("xelatex.exe -synctex=1 -interaction=nonstopmode report.tex")

```

Table S1. Descriptions of stratigraphic units at the Sheep Creek trench site

Unit ¹	Textural Description ²	Texture (%) ³				Clasts (cm)		Density/ Consistency	Cementation	Clast Angularity	Bedding/Structure	Sorting	Lower Boundary ⁴	Color (dry) ⁵	Notes
		F	S	G	C/B	Largest	Ave								
1	Gravelly cobbles with silt	15	5	35	45	35	5	Hard	Moderate	Sub-angular	Weak alluvial-fan bedding; channel imbrication	Poor	Not exposed	5Y 5/3	–
1Bk	Gravelly cobbles with Bk horizon	18	2	40	40	38	5	Slightly hard	Well	Sub-angular	Weak alluvial-fan bedding; channel imbrication	Poor	Gradual	10YR 6/4	Carbonate rinds on clasts
2	Silt with cobbles and gravel	40	<1	40	20	20	2	Slightly hard	Weak	Sub-angular	None; weak soil peds	Poor	Gradual	10YR 5/4	Locally vesicular texture
3	Gravelly silt with minor cobbles	70	<1	25	5	19	1–2	Slightly hard	Moderate	Sub-angular	None; blocky soil peds	Poor	Clear	10YR 4/3	Abundant roots from modern soil horizon
Cx	Silty gravel with cobbles	45	<1	40	15	40	5	Soft	Weak	Sub-angular	Weak slope-parallel bedding; blocky soil peds with modern soil	Poor	Clear	10YR 5/4	–
C2	Silty gravel with cobbles	35	<1	40	25	40	15	Soft	Moderate	Sub-angular	Weak slope-parallel bedding; blocky soil peds in soil CS2	Poor	Gradual	10YR 6/4	–
C1	Gravelly silt with cobbles	45	<1	35	20	30	5	Soft	Weak	Sub-angular	Weak slope-parallel bedding; blocky soil peds in soil C1S	Poor	Gradual	10YR 5/4	–
Unit descriptions recorded 17 September 2019. ¹ Unit names and locations correspond to supplementary Figure S2. ² Textural name approximated using the Unified Soil Classification System and Birkeland et al. (1991; Soils as a tool for applied Quaternary geology, Utah Geological and Mineralogical Survey, Miscellaneous Publication 91-3). Textural information may not be representative of entire unit due to vertical and horizontal heterogeneity in units. ³ Percentages of clast-size fractions (based on area) are field estimates. F – fines (silt and clay), S – sand, G – gravel, C/B – cobbles and boulders. A U.S. Standard #10 (2mm) sieve was used to separate matrix (clay, silt, and sand) from gravel. ⁴ Lower Boundary (modified from Birkeland et al., 1991). Distinctness: clear (2.5–6 cm), gradual (6–12.5 cm). ⁵ Munsell color of matrix (year 2000 - revised version).															

Table S2. Descriptions of stratigraphic units at the Arentson Gulch trench site

Unit; location ¹	Textural Description ²	Texture (%) ³				Clasts (cm)		Density/ Consistency	Cementation	Clast Angularity	Bedding/Structure	Sorting	Lower Boundary ⁴	Color (dry) ⁵	Notes
		F	S	G	C/B	Largest	Ave								
1; HW	Sandy gravel with minor silt	5	20	70	5	15	1–2	Slightly hard	Weak	Sub-angular	Moderate; slightly imbricated horizontal beds	Poor to moderate	Not exposed	2.5Y 5/2	Alluvial-fan channel gravel
1Bk; HW	Silty gravel with minor silt	15	10	65	10	17	2–3	Hard	Well, grading upward	Sub-angular	Poorly bedded; soil peds at upper contact	Poor	Gradual	2.5Y 6/3	
3; HW	Silt	90	4	5	1	15	1	Very hard	Moderate	Sub-angular	Massive	Poor	Clear	2.5Y 6/3	Carbonate root-cast stringers
4; HW	Silt	90	2	7	1	20	1	Hard	Moderate	Sub-angular	Massive	Poor	Gradual	10YR 5/4	Root casts, some carbonate stringers
4S; HW	Soil on silt	97	2	1	0	2	–	Hard	Moderate	–	Massive; small soil peds near upper contact	–	Gradual	10YR 4/3	Fan with soil separated by channel gravels. Includes modern soil.
5; HW	Silty gravel with buried soil	40	5	40	15	15	2.5	Soft	Weak	Sub-angular	Moderately bedded; gravel lenses	Poor to moderate	Clear	10YR 3/3	Fine carbonate laminations
2; HW	Laminated silt with minor gravel	88	5	5	2	10	2	Slightly hard	Moderate	Sub-angular	Fine carbonate laminations	Moderate	Abrupt	10YR 6/3	
C2; HW	Silty gravel with minor sand	50	10	30	10	18	~3	Hard	Well	Sub-angular	slope-parallel bedding/imbrication	Poor to moderate	Clear	2.5Y 6/4	Upward fining
C2S; HW	Silty gravel with minor sand	55	10	30	5	10	~2	Slightly hard	Moderate	–	Soil peds near top; carbonate laminations	–	Clear	2.5Y 5/4	Carbonate stringers
C1; HW	Silty gravel with minor sand	60	10	25	5	14	1–2	Slightly hard	Moderate	Sub-angular	Weak slope-parallel bedding	Poor to moderate	Gradual	10YR 5/3	Upward fining; some burrows; includes modern soil horizon
3; FW	Silt	90	4	5	1	25	1	Hard	Moderate	Sub-angular	Massive; soil peds; some vesicular texture	Poor	Gradual	10YR 5/3	Some possible rip-up clasts from unit below
4; FW	Silt with minor gravel	87	2	10	1	10	1	Slightly hard	Weak	Sub-angular	Massive; soil peds near top of unit	Poor	Gradual	7.5YR 4/4	Lacks carbonate root casts
Unit descriptions recorded 16 September 2019.															
¹ Unit names and locations correspond to supplementary Figures S1. HW – hanging wall; FW – footwall.															
² Textural name approximated using the Unified Soil Classification System and Birkeland et al. (1991; Soils as a tool for applied Quaternary geology, Utah Geological and Mineralogical Survey, Miscellaneous Publication 91-3). Textural information may not be representative of entire unit due to vertical and horizontal heterogeneity in units.															
³ Percentages of clast-size fractions (based on area) are field estimates. F – fines (silt and clay), S – sand, G – gravel, C/B – cobbles and boulders. A U.S. Standard #10 (2mm) sieve was used to separate matrix (clay, silt, and sand) from gravel.															
⁴ Lower Boundary (modified from Birkeland et al., 1991). Distinctness: abrupt (1 mm–2.5 cm), clear (2.5–6 cm), gradual (6–12.5 cm).															
⁵ Munsell color of matrix (year 2000 - revised version).															

Figure S1a. South wall of the Sheep Creek trench, Lost River Fault Zone

Figure S1b. North wall of the Sheep Creek trench, Lost River fault zone

Figure S2a. South Wall of the Arentson Gulch trench, Lost River Fault Zone

Figure S2b. North wall of the Arentson Gulch trench, Lost River fault zone

Figure S3.

Figure S4.

Figure S5.

S5a. Photograph of the Sheep Creek trench site; view is to the east. Orange flags mark area of trench excavation. 10 September 2019, by C. DuRoss.

S5e. Low-altitude photograph of the Sheep Creek trench; view is to the south. 17 September 2019, by Z. Lifton.

S5b. Photograph of the north wall of the Sheep Creek trench; view is to the north. Scale bar in photo is ~0.5 m long. 12 September 2019, by C. DuRoss.

S5c. Low-altitude photograph of the Sheep Creek trench. View is down and east. 17 September 2019, by Z. Lifton.

S5d. Low-altitude photograph of the Sheep Creek trench; view is to the east. 16 September 2019, by M. Bunds.

S5f. Photograph of the Arentson Gulch trench site; view is to the northeast. Orange flags mark area of trench excavation. 11 September 2019, by C. DuRoss.

S5j. Photograph of the Arentson Gulch trench site; view is to the southeast. 11 September 2019, by C. DuRoss.

S5g. Photograph of the Arentson Gulch trench site; view is down. 16 September 2019, by Z. Lifton.

S5h. Photograph of the Arentson Gulch trench site; view is northeast. 16 September 2019, by Z. Lifton.

S5i. Photograph of the Arentson Gulch trench site; view is to the northeast. 11 September 2019, by C. DuRoss.

Figure S6a.

Figure S6b.

Figure S6c.

Figure S6d.