Journal of Software for Algebra and Geometry

Seminormalization package for Macaulay2

KARL SCHWEDE AND BERNARD SERBINOWSKI

vol 10 2020

JSAG **10** (2020), 1–7
https://doi.org/10.2140/jsag.2020.10.1

The Journal of Software for
Algebra and Geometry

Seminormalization package for Macaulay2

KARL SCHWEDE AND BERNARD SERBINOWSKI

ABSTRACT: This note describes a package for computing seminormalization of rings within Macaulay2.

1. INTRODUCTION. Given a reduced excellent Noetherian ring R, between R and its normalization R^{N}, there is the seminormalization R^{SN}. In this paper we discuss an implementation of a seminormalization algorithm within Macaulay2. The ring R is called *seminormal* if every finite birational extension $R \subseteq S$ that induces a bijection on primes and induces isomorphisms, on the residue fields, is in fact an isomorphism; see for instance [Traverso 1970; Greco and Traverso 1980; Leahy and Vitulli 1981]. In particular, a *cusp* is not seminormal, since its normalization map is a bijection on points and induces isomorphisms of residue fields.

Let us delve a little deeper into non-normal rings, in a way that will help explain the algorithm. Suppose that R is as above with normalization R^{N}. The conductor $\mathfrak{c} \subseteq R \subseteq R^{\mathrm{N}}$ is defined to be $\mathrm{Ann}_R(R^{\mathrm{N}}/R)$. It is an ideal in both R and R^{N} which defines the locus where R is not normal. In this situation, R is always the pullback of the following diagram:

$$\begin{array}{ccc} R^{\mathrm{N}}/\mathfrak{c} & \hookleftarrow & R/\mathfrak{c} \\ \uparrow & & \uparrow \\ R^{\mathrm{N}} & \leftarrow & R \end{array} \tag{1}$$

Or in other words,

$$R \cong \{(x,\, y+\mathfrak{c}) \in R^{\mathrm{N}} \times R/\mathfrak{c} \mid x+\mathfrak{c} = y+\mathfrak{c}\}.$$

Since the pullback of this diagram dualizes to the pushout when taking Spec, we can interpret Spec R as a quotient of Spec R^{N} where certain points are identified (or have their residue fields shrunk) and certain tangent spaces are glued or otherwise annihilated (the latter owing to the *scheme structure* of $R^{\mathrm{N}}/\mathfrak{c}$). For additional discussion, see for instance [Schwede 2017]. In view of this construction, a ring is seminormal if its non-normality is due *only* to gluing of points. In other words, a seminormal ring is one where there is no undue identification of tangent spaces.

This idea leads us to our algorithm for seminormalizing, which is the topic of the next section.

Schwede was supported in part by NSF CAREER Grant DMS #1252860/1501102 and NSF grant #1801849. Serbinowski was supported in part by NSF CAREER Grant DMS #1252860/1501102.
MSC2010: primary 13F45; secondary 13B22, 14M05.
Keywords: seminormalization, normalization, Macaulay2.
`Seminormalization` version 0.21

2. STRUCTURE OF THE ALGORITHM. The idea of the algorithm is to perform the pullback from (1). However, instead of just forming $R/\mathfrak{c}$ and $R^{\mathrm{N}}/\mathfrak{c}$, we want to remove unnecessary tangent space identification. A simple option would be to form the pullback S of the diagram

$$\begin{array}{ccc} R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}} & \hookleftarrow & R/\sqrt{\mathfrak{c}R} \\ \uparrow\uparrow & & \uparrow \\ R^{\mathrm{N}} & \leftarrow\cdots & S \end{array} \tag{2}$$

but this is not the seminormalization of R since $R/\sqrt{\mathfrak{c}R}$ could itself have undue gluing of tangent spaces. An easy way to get around this is to seminormalize $R/\sqrt{\mathfrak{c}R}$ (which has lower dimension than R, and so a recursive algorithm can apply), but the seminormalization $(R/\sqrt{\mathfrak{c}R})^{\mathrm{SN}}$ does not necessarily map to $R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}}$ (since that is not necessarily seminormal). We could also seminormalize $R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}}$, but this led to implementation difficulties and so we instead form the intersection

$$D = (R/\sqrt{\mathfrak{c}R})^{\mathrm{SN}} \cap (R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}}),$$

where the intersection takes place in the total ring of fractions of $R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}}$. Then we pullback the diagram

$$\begin{array}{ccc} R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}} & \hookleftarrow & D \\ \uparrow\uparrow & & \uparrow \\ R^{\mathrm{N}} & \leftarrow\cdots & C \end{array}$$

Theorem 2.1. *The ring C in the above diagram is the seminormalization of R.*

Proof. We first notice that there is a diagram

$$\begin{array}{ccc} (R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}})^{\mathrm{SN}} & \hookleftarrow & (R/\sqrt{\mathfrak{c}R})^{\mathrm{SN}} \\ \uparrow & & \uparrow{\scriptstyle \beta} \\ R^{\mathrm{N}} & \leftarrow\cdots & R^{\mathrm{SN}} \end{array}$$

coming from the functoriality of seminormalization. Since the image of R^{SN} maps into $(R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}})$, we see that $\beta(R^{\mathrm{SN}}) \subseteq D$. Hence by the universal property of pullback, there is a map $R^{\mathrm{SN}} \to C$. On the other hand, it follows from [Ferrand 2003] (cf. [Schwede 2005]), that the map $R^{\mathrm{SN}} \to C$ induces a bijection on points of Spec and induces an isomorphism of residue fields. Indeed, recall that the pullback of a diagram $\{A \to A/I \leftarrow B\}$ replaces the closed subscheme $V(I) \subseteq \operatorname{Spec} A$ with Spec B, residue fields and all. Since $C \hookrightarrow R^{\mathrm{N}}$, we have that $R^{\mathrm{SN}} \to C$ is also birational and so $R^{\mathrm{SN}} \to C$ is an isomorphism since R^{SN} is seminormal by definition. This completes the proof. □

3. IMPLEMENTATION OF THE ALGORITHM. We describe the main algorithm first and then describe the strategy of some of the component functions individually.

3.1. ***The main seminormalization algorithm.*** As the algorithm is recursive, the first thing we do is check whether we are finished. If the Krull dimension of the ring is 0 or if the ring is normal we return the ring unchanged as it has already been seminormalized (note we assume that the ring the function is given is reduced). This ensures that the process will end since at each step of the induction, the dimension will drop.

Assuming the ring is not already seminormal, then we create a map from the input ring to its normalization:

$$\phi : R \to R^{\mathrm{N}}.$$

Note, we do not use the core normalization function `integralClosure` as that does not work correctly on nondomains (even if the ring is reduced). Instead we call a function `betterNormalizationMap` (which eventually calls `integralClosure` on various quotient rings of R); see Section 3.2.

We then compute the conductor of this map, which we continue to call $\mathfrak{c}$. We then take the radical of this ideal in both R and R^{N} and now we can form the diagram

$$\begin{array}{ccc} R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}} & \hookleftarrow & R/\sqrt{\mathfrak{c}R} \\ \phi \uparrow\uparrow & & \\ R^{\mathrm{N}} & & \end{array}$$

At this point, we make our recursive call and seminormalize $R/\sqrt{\mathfrak{c}R}$ (constructing a map $\gamma : R/\sqrt{\mathfrak{c}R} \to R/\sqrt{\mathfrak{c}R}^{\mathrm{SN}}$ in the process). Finally, we need to construct the ring we called D above,

$$D = (R/\sqrt{\mathfrak{c}R})^{\mathrm{SN}} \cap R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}}.$$

This is a bit tricky, and is the subject of Section 3.3 below. In the meantime, once we have constructed D (and the map $\psi : D \to R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}}$), we can pullback the diagram

$$\begin{array}{ccc} R^{\mathrm{N}}/\sqrt{\mathfrak{c}R^{\mathrm{N}}} & \overset{\psi}{\hookleftarrow} & D \\ \phi \uparrow\uparrow & & \uparrow \\ R^{\mathrm{N}} & \leftarrow\cdots & C \end{array}$$

to obtain C, which we have already verified is the seminormalization in Theorem 2.1. Note we perform this pullback by using the package [Pullback], which requires the map ϕ above to be surjective.

3.2. ***Normalization of reduced rings.*** As mentioned above, a function `betterNormalizationMap`, which computes the normalization of reduced rings is included in this package. The strategy is as follows.

Step 1: Compute the minimal primes $\{\mathfrak{q}_i\}$ of R.

Step 2: Compute the normalizations of the $R/\mathfrak{q}_i$. Note the function `betterNormalizationMap` has an option `Strategy` which is passed to the `integralClosure` calls at this step.

Step 3: Construct the product of the normalized rings $R^{\mathrm{N}} = \prod_i (R/\mathfrak{q}_i)^{\mathrm{N}}$.

Step 4: Construct the map from R to R^{N}.

Step 3, constructing the product of normalized rings, is achieved by calling a function `ringProduct` which computes a product of a list of rings defined over the same base ring (i.e., defined over `QQ`). This returns the product of rings as well as the list of orthogonal idempotents defining each ring. It also returns a list of lists showing what variables from our original rings become in the product. We hope that this functionality of taking products of rings may be useful in other contexts besides computing normalizations.

Step 4 is the most involved. We first construct various maps $(R/\mathfrak{q}_i)^{\mathrm{N}} \to R^{\mathrm{N}}$. Notice, this is not a *real* ring map which we want to study; we are using it to keep track of where variables of the rings $(R/\mathfrak{q}_i)^{\mathrm{N}}$ go. We then compose with $R \to (R/\mathfrak{q}_i)^{\mathrm{N}}$ to obtain various different maps $R \to R^{\mathrm{N}}$. Finally, we sum these maps (multiplying by our orthogonal idempotents as appropriate) to obtain our normalization map $R \to R^{\mathrm{N}}$.

3.3. ***Intersecting the seminormalization and another ring extension.*** At a key point in our algorithm, we have two extensions of $A = R/\sqrt{\mathfrak{c}R}$, first the seminormalization $A^{\mathrm{SN}} = (R/\sqrt{\mathfrak{c}R})^{\mathrm{SN}}$ and second, the finite extension to $B = R^N/\sqrt{\mathfrak{c}R^N}$. We need to form an intersection of these two extensions. We do this by using the function `intersectSeminormalizationAndExtension` which computes exactly this intersection (and a ring map from our base ring to the intersection).

To do this, first we find a reduced ring O containing both of these extensions whose minimal primes are in bijection with the minimal primes of the ring we called B. This is done via the function `findOverring` which essentially tensors the two extension rings together and then drops any unnecessary minimal primes. Note we do not have to worry about how the seminormalization embeds into this overring by uniqueness properties of elements of the seminormalization [Swan 1980]. Once we have the overring O, we form the exact sequence (making liberal use of the `pushFwd` function in the [PushForward] package)

$$A^{\mathrm{SN}} \oplus B \xrightarrow{(a,b)\mapsto a-b} O \to 0$$

and computing the kernel K. At this point, K is the desired intersection ring, but Macaulay2 only understands it as a module. However, we can take the module generators of K, map them into B, and consider the ring they generate. This is our desired ring.

3.4. ***Variable naming conventions.*** One issue we ran into when calling a recursive function that produces new rings is that there can be numerous collisions of variable names which makes debugging very difficult. Because of this, internally, there is a complicated scheme for naming variables.

However, none of this is visible in the outputted ring, as by default *all the variables* of the output ring will have been renamed as $\mathtt{Yy}_N$ where N varies. If you do not want to use `Yy`, you may instead supply your own variable name via the `Variable ⇒ X` option when calling `seminormalize`. Here `X` must be a valid symbol. It is important to note that you cannot use a symbol that overlaps with an existing variable as this will cause errors. The output does include a map from the original ring to the seminormalization.

4. EXAMPLES. The function `seminormalize` returns a list of three things. First it returns the ring R^{SN}, then it returns the ring map $R \to R^{\mathrm{SN}}$ and finally it returns the ring map $R^{\mathrm{SN}} \to R^{\mathrm{N}}$.

We begin by seminormalizing the cusp; in this case the seminormalization is the normalization.

```
i1 : loadPackage "Seminormalization"
o1 = Seminormalization
o1 : Package
i2 : R = QQ[x,y]/ideal(y^2-x^3);
i3 : seminormalizedList = (seminormalize(R));
i4 : seminormalizedList#0

                QQ[Yy , Yy , Yy ]
                      0   1   2
o4 = ---------------------------------------
       2                     2
     (Yy  - Yy , Yy Yy  - Yy , Yy  - Yy Yy )
       2     1    1  2     0    1     0  2

o4 : QuotientRing
i5 : prune seminormalizedList#0
o5 = QQ[Yy ]field
          2
o5 : PolynomialRing
i6 : seminormalizedList#1

                    QQ[Yy , Yy , Yy ]
                          0   1   2
o6 = map(---------------------------------------,R,{Yy , Yy })
           2                     2                  1   0
         (Yy  - Yy , Yy Yy  - Yy , Yy  - Yy Yy )
           2     1    1  2     0    1     0  2

                        QQ[Yy , Yy , Yy ]
                              0   1   2
o6 : RingMap ---------------------------------------- <--- R
               2                     2
             (Yy  - Yy , Yy Yy  - Yy , Yy  - Yy Yy )
               2     1    1  2     0    1     0  2

i7 : isSeminormal(R)
o7 = false
```

Next, we seminormalize the union of four lines through the origin in $\mathbb{A}^2$. This should produce something isomorphic to the union of the 4 coordinate axes in $\mathbb{A}^4$, which it does.

```
i2 : R = QQ[x,y]/ideal(x*y*(x^2-y^2));
i3 : seminormalizedList = seminormalize(R);
i4 : seminormalizedList#0

                  QQ[Yy , Yy , Yy , Yy ]
                        0   1   2   3
o4 = -------------------------------------------------
     (Yy Yy , Yy Yy , Yy Yy , Yy Yy , Yy Yy , Yy Yy )
        2  3    1  3    0  3    1  2    0  2    0  1

o4 : QuotientRing
```

The following example of Greco and Traverso [1980] is a seminormal ring whose prime spectrum has an irreducible component that is not seminormal.

```
i2 : B = ZZ/11[x,y,u,v,e,f];
i3 : I = intersect(ideal(u,v,e-1,f),ideal(x,y,e,f-1));
i4 : A = B/I;
i5 : E = ZZ/11[z1, z2, z3, z4, z5];
i6 : h = map(A, E, {x^3+u, x^2+v, y, u^2-v^3, x*y});
i7 : J = ker h;
i8 : D = E/J;
i9 : isSeminormal(D) --this should be seminormal
o9 = true
i10 : JJ = preimage(h, ideal(sub(f,A)));
i11 : D2 = E/(trim(JJ + J));
i12 : isSeminormal(D2) --this should not be seminormal
o12 = false
```

Finally, we verify the seminormality of a seminormal ring that is not weakly normal.

```
i2 : R = ZZ/2[t, x, y]/ideal(x^2 - t*y^2);
i3 : isSeminormal(R)
o3 = true
```

5. FUTURE WORK. There are a number of ways that this package could be improved in the future. We list some of them here in the hope that they will inspire others, and remind us, to work on them.

(i) Implement this algorithm over more general coefficient rings.

(ii) Implement weak normalization, if possible.

(iii) Implement functorial seminormalization (in other words, given a map between two rings, there should always be a unique map between their seminormalizations).

(iv) Improve the speed of the computation where possible.

ACKNOWLEDGEMENTS. The authors thank Neil Epstein and Claudiu Raicu for stimulating discussions and in particular Claudiu Raicu for writing and then improving the [PushForward] package in ways that helped the development of this package. We also thank the referees for numerous valuable comments and suggestions, both on the paper and on the package.

SUPPLEMENT. The online supplement contains version 0.21 of `Seminormalization`.

REFERENCES.

[Ferrand 2003] D. Ferrand, "Conducteur, descente et pincement", *Bull. Soc. Math. France* **131**:4 (2003), 553–585. MR

[Greco and Traverso 1980] S. Greco and C. Traverso, "On seminormal schemes", *Compositio Math.* **40**:3 (1980), 325–365. MR Zbl

[Leahy and Vitulli 1981] J. V. Leahy and M. A. Vitulli, "Seminormal rings and weakly normal varieties", *Nagoya Math. J.* **82** (1981), 27–56. MR Zbl

[Pullback] D. Ellingson and K. Schwede, "Pullback: pullback of rings", Macaulay2 package, version 1.03, available at https://github.com/Macaulay2/M2/blob/master/M2/Macaulay2/packages/Pullback.m2.

[PushForward] C. Raicu, "PushForward: push forwards of finite ring maps", Macaulay2 package, version 0.1, available at https://github.com/Macaulay2/M2/tree/master/M2/Macaulay2/packages.

[Schwede 2005] K. Schwede, "Gluing schemes and a scheme without closed points", pp. 157–172 in *Recent progress in arithmetic and algebraic geometry*, edited by F. Bacchus and T. Walsh, Contemp. Math. **386**, Amer. Math. Soc., Providence, RI, 2005. MR Zbl

[Schwede 2017] K. Schwede, Reply to "Is there a geometric intuition underlying the notion of normal varieties?", MathOverflow, 2017, available at http://mathoverflow.net/q/109486.

[Swan 1980] R. G. Swan, "On seminormality", *J. Algebra* **67**:1 (1980), 210–229. MR Zbl

[Traverso 1970] C. Traverso, "Seminormality and Picard group", *Ann. Scuola Norm. Sup. Pisa* (3) **24**:4 (1970), 585–595. MR Zbl

RECEIVED: 16 Oct 2018 REVISED: 14 Oct 2019 ACCEPTED: 5 Dec 2019

KARL SCHWEDE:

schwede@math.utah.edu
Department of Mathematics, The University of Utah, Salt Lake City, UT, United States

BERNARD SERBINOWSKI:

bserbinowski@gmail.com
Department of Computer Science, Vanderbilt, Nashville, TN, United States