Generic curve with
module basis size different from
the number of ring generators

1 Example

Consider the generic type I example

$$A := \overline{\mathbf{F}}[f_{11}, f_6]/\langle (f_{11}^2 - f_{11})^3 - (f_{11}^2 - f_{11})f_{11}f_6^4 - f_6^{11}\rangle$$

made up to have a non-trivial integral closure, since $f_{10} := (y^2 - y)/x^2$ is clearly in it, with weight 10, less than the weight of f_{11}. It also rated to have few, if any, coefficients to reconstruct over the rationals, and should have been small enough to be easy for all implementations in any characteristic.

2 qth-power algorithm

My qth-power algorithm code in Magma in characteristic 0 would compute results for $\mathbf{F}_3$, $\mathbf{F}_5$ and $\mathbf{F}_7$, piecing them together to get results for $\mathbf{Q}$ and $\mathbf{F}_p$ for most primes p.

This gives a strict $P := \mathbf{F}[f_6]$-algebra presentation with basis having weights $(0, 10, 11, 13, 15, 20)$, from which the genus $g = 9$ can be inferred directly as the number of missing pole orders $(1, 2, 3, 4, 5, 7, 8, 9, 14)$.

```
Loading file "qchar051009.mag"
Loading "closure_func040909.mag"
1111111111111111111111111111111111111111111111111111111
q= 3
Delta= P.2^11
WT_MATRIX_T= [
    [ 20, 15, 13, 11, 10, 6 ]
]
time for q= 3 is 0.050 seconds
modulus= 3
1111111111111111111111111111111111111111111111111111111
q= 5
Delta= P.2^11
WT_MATRIX_T= [
    [ 20, 15, 13, 11, 10, 6 ]
]
time for q= 5 is 0.090 seconds
modulus= 15
[
    f_11^4*f_6^3 + f_11^3*f_6^3 + f_11^2*f_6^3,
    f_11^3*f_6^4 + f_11^2*f_6^4 + f_11*f_6^4,
```

```
    f_11^5 - f_11^2*f_6^4 + f_11*f_6^4 - f_11^2,
    f_11*f_6^7,
    f_11^2*f_6^5 - f_11*f_6^5,
    f_6^8,
    f_6^7
]
1111111111111111111111111111111111111111111111111111111111
q= 7
Delta= P.2^11
WT_MATRIX_T= [
    [ 20, 15, 13, 11, 10, 6 ]
]
time for q= 7 is 0.110 seconds
modulus= 105
[
    f_11^4*f_6^3 - 2*f_11^3*f_6^3 + f_11^2*f_6^3,
    f_11^3*f_6^4 - 2*f_11^2*f_6^4 + f_11*f_6^4,
    f_11^5 - f_11^2*f_6^4 - 3*f_11^4 + f_11*f_6^4 + 3*f_11^3 - f_11^2,
    f_11*f_6^7,
    f_11^2*f_6^5 - f_11*f_6^5,
    f_6^8,
    f_6^7
]
1 f_11^4*f_6^3 - 2*f_11^3*f_6^3 + f_11^2*f_6^3
2 f_11^3*f_6^4 - 2*f_11^2*f_6^4 + f_11*f_6^4
3 f_11^5 - f_11^2*f_6^4 - 3*f_11^4 + f_11*f_6^4 + 3*f_11^3 - f_11^2
4 f_11*f_6^7
5 f_11^2*f_6^5 - f_11*f_6^5
6 f_6^8
7 f_6^7

newrelations= [
    f_10^2 - f_20,
    f_11^2 - f_10*f_6^2 - f_11,
    f_11*f_10 - f_15*f_6 - f_10,
    f_13^2 + f_20*f_6 - f_11*f_6 - f_15 + f_6,
    f_13*f_11 + f_6^4,
    f_13*f_10 + f_11*f_6^2 - f_6^2,
    f_15^2 + f_6^5 - f_15*f_6 + f_13*f_6,
    f_15*f_13 + f_10*f_6^3 - f_11*f_6 + f_6,
    f_15*f_11 - f_20*f_6,
    f_15*f_10 - f_13*f_6^2 - f_10*f_6,
    f_20^2 + f_10*f_6^5 - f_13*f_6^3 - f_10*f_6^2 - f_20,
    f_20*f_15 + f_11*f_6^4 - f_20*f_6 - f_6^4,
    f_20*f_13 + f_15*f_6^3,
    f_20*f_11 - f_13*f_6^3 - f_10*f_6^2 - f_20,
```

```
    f_20*f_10 + f_6^5 - f_15*f_6 - f_10
]
totaltime= 0.280 seconds
```

3 Singular

Singular's `normal` function in characteristic 0 gives

```
                     SINGULAR                                 /
 A Computer Algebra System for Polynomial Computations       /   version 3-1-3
                                                           0<
 by: W. Decker, G.-M. Greuel, G. Pfister, H. Schoenemann     \   March 2011
FB Mathematik der Universitaet, D-67653 Kaiserslautern        \
> LIB "normal.lib";
> ring r=0,(f11,f6),wp(11,6);
> ideal i=(f11^2-f11)^3-(f11^2-f11)*f11*f6^4-f6^11;
> list nor=normal(i);nor;
//   characteristic : 0
//   number of vars : 7
//        block   1 : ordering dp
//                  : names    T(1) T(2) T(3) T(4) T(5)
//        block   2 : ordering wp
//                  : names    f11 f6
//                  : weights  11  6
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=3*f11*f6^10+3*f11^4*f6^3+3*f6^10-2*f11^3*f6^3-f11^2*f6^3
      _[2]=3*f11^4*f6^4-3*f11^3*f6^4-2*f6^8-3*f11^2*f6^4+3*f11*f6^4
      _[3]=3*f11*f6^9+3*f11^4*f6^2+3*f6^9-2*f11^3*f6^2-f11^2*f6^2
      _[4]=3*f11^2*f6^8-2*f11^4*f6-3*f6^8+2*f11*f6^5+4*f11^3*f6-2*f11^2*f6
      _[5]=9*f11^3*f6^7-9*f11*f6^7+6*f11^2*f6^4+6*f11^4-4*f11*f6^4-12*f11^3+6*f11^2
      _[6]=3*f11^5-3*f11^4-2*f11*f6^4-3*f11^3+3*f11^2
> def R=nor[1][1];
> setring R;
> option(redSB);
> ideal s=std(norid);s;
s[1]=f11^6-f6^11-f11^3*f6^4-3*f11^5+f11^2*f6^4+3*f11^4-f11^3
s[2]=T(5)*f11^2-T(5)*f11-3*f6^7-2*f11^2+2*f11
s[3]=T(5)*f6^4-3*f11^4+3*f11*f6^4+6*f11^3-2*f6^4-3*f11^2
s[4]=3*T(4)*f11-T(5)*f6+2*f6
s[5]=T(4)*f6^3-f11^3+f6^4+2*f11^2-f11
s[6]=T(3)*f11-T(3)-T(4)*f6-f6^2
s[7]=T(3)*f6^2-f11^2+f11
s[8]=T(2)*f11-f6^4
s[9]=3*T(2)*f6^3-T(5)*f11+T(5)+2*f11-2
s[10]=T(1)-T(3)*f6
s[11]=T(5)^2+3*T(5)*f11-4*T(5)-9*f11^2*f6^3+9*f11*f6^3-6*f11+4
s[12]=T(4)*T(5)-2*T(4)+T(5)*f6-3*f11*f6^4+3*f6^4-2*f6
s[13]=T(3)*T(5)-2*T(3)-3*f6^5
```

```
s[14]=T(2)*T(5)-2*T(2)-3*f11^3+3*f6^4+6*f11^2-3*f11
s[15]=T(4)^2+T(2)*f6+T(4)*f6-f6^5
s[16]=T(3)*T(4)-T(2)*f6^2
s[17]=T(2)*T(4)+T(2)*f6-f11^2*f6+2*f11*f6-f6
s[18]=3*T(3)^2-T(5)-3*f11+2
s[19]=T(2)*T(3)-f11*f6^2+f6^2
s[20]=3*T(2)^2+3*T(4)-T(5)*f6+2*f6
```

From s[10], we may infer that T(1) is totally unnecessary. [There is no excuse for this, as a simple interreduction before post-processing would remove this and possibly simplify the rest. Even failing this it is easy to remove such in the same way (but for good reason) that `normalP` removes variables (for no particular reason except that it can).] From the fractions or from s[2], s[4], s[6], and s[8], we may infer that $wt(T(5)) = 20$, $wt(T(4)) = 15$, $wt(T(3)) = 10$, and $wt(T(2)) = 13$ respectively, though clearly `normal` doesn't try to assign weights, but uses a generic product monomial ordering, grevlex-over-input, instead.

Note that by rewriting each new fraction with denominator f_6^7, it is possible to reduce the numerators: for $T(5)$ to that for $3f_{20} - 3f_{11} + 2$, for $T(4)$ to that for $f_{15} - f_6$, for $T(3)$ to that for f_{10} and for $T(2)$ to that for f_{13}. This is enough to guarantee that this is isomorphic to the presentation I gave above. Checking it by reducing my answers relative to these fractions is done by ?????

Now consider `normalP` with and without the `noRed` option:

```
> ring r=5,(f11,f6),wp(11,6);
> ideal i=(f11^2-f11)^3-(f11^2-f11)*f11*f6^4-f6^11;
> list norp=normalP(i,"withRing","noRed");norp;
//   characteristic : 5
//   number of vars : 6
//        block   1 : ordering dp
//                  : names    T(1) T(2) T(3) T(4)
//        block   2 : ordering wp
//                  : names    f11 f6
//                  : weights  11  6
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=f6^11+f11^3*f6^4+f11^5-f11^2*f6^4-2*f11^4+f11^3
      _[2]=f11^3*f6^6-f11^2*f6^6
      _[3]=f11*f6^10
      _[4]=f11^2*f6^9
      _[5]=f11^4*f6^2-f11^3*f6^2
[3]:
   [1]:
      16
   [2]:
      16
> def Rp=norp[1][1];
> setring Rp;
> option(redSB);
> ideal sp=std(norid);sp;
sp[1]=f11^6-f6^11-f11^3*f6^4+2*f11^5+f11^2*f6^4-2*f11^4-f11^3
sp[2]=T(4)*f11^2-T(4)*f11-f6^7
sp[3]=T(4)*f6^4-f11^4+f11*f6^4+2*f11^3-f11^2
sp[4]=T(3)*f11-T(4)*f6
sp[5]=T(3)*f6^3-f11^3+f6^4+2*f11^2-f11
sp[6]=T(2)*f11-f6^4
sp[7]=T(2)*f6^3-T(4)*f11+T(4)
sp[8]=T(1)*f11-T(1)-T(3)*f6-f6^2
sp[9]=T(1)*f6^2-f11^2+f11
sp[10]=T(4)^2+T(4)*f11-f11^2*f6^3+f11*f6^3
sp[11]=T(3)*T(4)+T(4)*f6-f11*f6^4+f6^4
sp[12]=T(2)*T(4)-f11^3+f6^4+2*f11^2-f11
sp[13]=T(1)*T(4)-f6^5
sp[14]=T(3)^2+T(2)*f6+T(3)*f6-f6^5
sp[15]=T(2)*T(3)+T(2)*f6-f11^2*f6+2*f11*f6-f6
sp[16]=T(1)*T(3)-T(2)*f6^2
sp[17]=T(2)^2+T(3)-T(4)*f6
```

```
sp[18]=T(1)*T(2)-f11*f6^2+f6^2
sp[19]=T(1)^2-T(4)-f11
```

With `noRed` one expects an answer similar to that givne by `normal`; but here the extraneous function of weight 16 is missing and the functions happen to come out in order of their weights, meaning $wt(T(1)) = 10$, $wt(T(2)) = 13$, $wt(T(3)) = 15$, and $wt(T(4)) = 20$.

```
//   characteristic : 5
//   number of vars : 4
//        block   1 : ordering dp
//                  : names    T(1) T(2)
//        block   2 : ordering wp
//                  : names    f11 f6
//                  : weights  11  6
//        block   3 : ordering C
[2]:
   [1]:
      _[1]=f6^11+f11^3*f6^4+f11^5-f11^2*f6^4-2*f11^4+f11^3
      _[2]=f11^3*f6^6-f11^2*f6^6
      _[3]=f11*f6^10
      _[4]=f11^2*f6^9
      _[5]=f11^4*f6^2-f11^3*f6^2
[3]:
   [1]:
      16
   [2]:
      16
> def Rp=norp[1][1];
> setring Rp;
> option(redSB);
> ideal sp=std(norid);sp;
sp[1]=f11^6-f6^11-f11^3*f6^4+2*f11^5+f11^2*f6^4-2*f11^4-f11^3
sp[2]=T(2)*f11-f6^4
sp[3]=T(2)*f6^7-f11^5+f11^2*f6^4-2*f11^4-f11*f6^4+2*f11^3+f11^2
sp[4]=T(1)*f6^2-f11^2+f11
sp[5]=T(1)*f11^3-2*T(1)*f11^2+T(1)*f11-T(2)*f6^5-f11^2*f6^2+f11*f6^2
sp[6]=T(2)^2*f6-T(1)*f11^2+2*T(1)*f11-T(1)+f11*f6^2-f6^2
sp[7]=T(1)*T(2)-f11*f6^2+f6^2
sp[8]=T(1)^2*f11-T(1)^2-T(2)*f6^3-f11^2+f11
sp[9]=T(2)^3-T(2)*f6-f11^3*f6+f6^5-2*f11^2*f6+2*f11*f6+f6
sp[10]=T(1)^3-T(1)*f11-f6^5
```

Without `noRed`, `normalP` looks for variables showing up alone to eliminate. So from sp[19] and sp[17] above, we could remove T(4) and T(3) respectively. It probably does this from `norid` directly, which could possibly give a different result than doing it from `std(norid)`, though I have no evidence of this. The point is that this elimination is not really driven by good mathematics, but rather by the desire to eliminate variables somehow. More importantly, it is inconsistent with the theory set forth in chapter 3 of the SINGULAR book, namely that the presentation should be that of a strict affine A-algebra (strict being my description of a presentation with Gröbner basis having relations that are at most quadratic (here over A)). And now isomorphism in one direction is more complicated because of the functions that are now missing.

The reason this was done in characteristic 5 here is that `normalP` timed out starting at characteristic 7 on this example. If I interrupt it, it will invariably be at the command

```
K = preimage(Q,phi,L);    //### Improvement by block ordering?'
```

which is a computationally mindless implementation of the crucial step to compute $NormalForm(f^q, I)$ in which f^q is computed first before reduction mod I. [It is perhaps surprising that this fails at characteristic as small as 7, but in general one is producing an f^q that is reasonably large before trying to reduce it, rather than attempting a square and reduce implementation as is done routinely elsewhere mathematics to keep intermediate results small.]

4 Macaulay2

```
Macaulay2, version 1.4
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : loadPackage "QthPower";

i2 : wtr=matrix{{11,6}};

i3 : R=QQ[f11,f6,Weights=>entries weightGrevlex(wtr)];

i4 : I={(f11^2-f11)^3-(f11^2-f11)*f11*f6^4-f6^11};

i5 : A=R/ideal(I);

i6 : time icf=icFractions A;
     -- used 0.527341 seconds

i7 : toString icf

o7 = {(f6^8+f11^2*f6^4+f11^3*f6-2*f11*f6^4-f11^2*f6+f6^4)
     /(f6^4+f11^2-f11),
     (f11^3-2*f11^2+f11)
     /(f6^3),
      (f11^2-f11)
     /(f6^2),
       f11,
       f6}

i8 : G=transpose gens gb presentation integralClosure A

o8 = {-11} | f11^6-f6^11-f11^3f6^4-3f11^5+f11^2f6^4+3f11^4-f11^3
     {-2}  | w_(3,0)f6^2-f11^2+f11
     {-9}  | w_(3,0)f11^4-2w_(3,0)f11^3+w_(3,0)f11^2-f6^9-f11^3f6^2+f11^2f6^2
     {-7}  | w_(3,0)^2f11^2-w_(3,0)^2f11-f6^7-f11^3+f11^2
     {-5}  | w_(3,0)^3-w_(3,0)f11-f6^5
     {-1}  | w_(5,0)f6-w_(3,0)f11+w_(3,0)
     {-1}  | w_(5,0)f11-w_(3,0)^2f6
     {-5}  | w_(5,0)w_(3,0)^2-w_(3,0)^2f6-f11f6^4+f6^4
     {-5}  | w_(5,0)^2w_(3,0)-w_(3,0)^2f11+w_(3,0)^2-f11^2f6^3+2f11f6^3-f6^3
     {-6}  | w_(5,0)^3+w_(5,0)w_(3,0)-w_(3,0)f11f6-f11^3f6^2-f6^6+3f11^2f6^2-3f11f6^2+f6
     {0}   | w_(6,0)f6-w_(5,0)^2
     {-5}  | w_(6,0)f11-w_(3,0)^2f6-f11f6^4+f6^4
     {-4}  | w_(6,0)w_(3,0)-w_(5,0)w_(3,0)-f11^2f6^2+2f11f6^2-f6^2
     {-4}  | w_(6,0)w_(5,0)-w_(5,0)^2-f11^3f6+3f11^2f6-3f11f6+f6
```

```
{-8}  | w_(6,0)^2-w_(5,0)^2+w_(5,0)+2w_(3,0)^2f11f6-3w_(3,0)^2f6-f6^8-2f11^3f6+4f11
-----------------------------------------------------------------------------
```

The implicit weights here are $wt(w_{3,0}) = 10$, $wt(w_{5,0}) = 15$, and $wt(w_{6,0}) = 24$. [It is not hard in this example to see that $w_{6,0} - f_6^4 - w_{5,0} = -f_6^4/f_{11}$ has weight 13.] This is not a strict presentation, and does not even use a great set of ring generators, though it is a technically correct answer.

But, absent the weights, how does one see that this is isomorphic the what was produced above?

```
R=ZZ/5[f11,f6,Weights=>entries weightGrevlex(wtr)];

I={(f11^2-f11)^3-(f11^2-f11)*f11*f6^4-f6^11};

A=R/ideal(I);

time icp=icFracP A;
     -- used 2.43643 seconds
toString icp

o16 = {1,
       (f6^4)
      /(f11),
       (f6^7-2*f11^3+f6^4-f11^2-2*f11)
      /(f6^3),
       (f11^2-f11)
      /(f6^2),
       (f11^3+2*f6^4-2*f11^2+f11)
      /(f6^3),
       (f6^7-2*f11^3-f11^2-2*f11)
      /(f11^2-f11)}
```

`icFracP` produces no presentation and fractions of implicit weights $0, 13, 24, 10, 15, 20$. It seems that

```
icp#2-f6^4+2*icp#4=0
```

in extraneous.

At characteristic as low as 11 this has problems. Interrupting it, we see that $K = intersect(kernel f, U)$; is the culprit; as with `normalP` above, being a computationally poor implementation of the crucial $NormalForm(f^q, I)$ step.

5 Magma

First try Magma's `Normalisation` function.

```
F:=Rationals();
P<f11,f6>:=PolynomialRing(F,2,"grevlexw",[11,6]);
I:=ideal<P|(f11^2-f11)^3-(f11^2-f11)*f11*f6^4-f6^11>;
time N:=Normalisation(I);
f11@N[1][2];
f6@N[1][2];
time G:=GroebnerBasis(N[1][1]);
G;
```

produced for me 4146 lines of output, probably with variables of weights $6, 10, 11, 13, 15$, using a lex monomial ordering. Compare this with an enlightened change of monomial ordering:

```
R<g15,g13,g11,g10,g6>:=PolynomialRing(F,5,"grevlexw",[15,13,11,10,6]);
phi:=hom<Parent(G[1])->R|g6,g10,g11,g13,g15>;
J:=phi(G);
time B:=GroebnerBasis(J);
B;

time: 0.980
[
    g15^2 - g6^5 - g15*g6 + g13*g6,
    g10^3 - g6^5 - g15*g6 - g10,
    g15*g13 - g10*g6^3 + g11*g6 - g6,
    g13^2 - g10^2*g6 + g11*g6 + g15 - g6,
    g15*g11 - g10^2*g6,
    g15*g10 - g13*g6^2 - g10*g6,
    g13*g11 - g6^4,
    g13*g10 - g11*g6^2 + g6^2,
    g11^2 - g10*g6^2 - g11,
    g11*g10 - g15*g6 - g10
]
```

This only needs a new variable $g20 := g10^2$ to make it into a strict $P := \mathbf{Q}[g6]$-algebra presentation.

`IntegralClosure` will work on this since there is one dependent and one independent variable. It gives

```
1 1
2 f11
3 1/f6^2*f11^2-1/f6^2*f11
4 1/f6^3*f11^3-2/f6^3*f11^2+1/f6^3*f11
5 1/f6^4*f11^4-2/f6^4*f11^3+1/f6^4*f11^2
6 1/f6^7*f11^5-3/f6^7*f11^4+3/f6^7*f11^3+(-f6^4-1)/f6^7*f11^2+1/f6^3*f11
```

over the rationals, but as always, no "stand-alone" presentation, no way of weighting the answer, and it is fortunate that the module basis it produced has good weights (that happen to be the weights of the left-most terms).

6 Qth-power

My `QthPower` package in Macaulay2 gives

```
Macaulay2, version 1.4
with packages: ConwayPolynomials, Elimination, IntegralClosure, LLLBases,
               PrimaryDecomposition, ReesAlgebra, TangentCone

i1 : loadPackage "QthPower";

i2 : wtr=matrix{{11,6}};

i3 : R=ZZ/11[f11,f6,Weights=>entries weightGrevlex(wtr)];

i4 : I={(f11^2-f11)^3-(f11^2-f11)*f11*f6^4-f6^11};

i5 : time ic=rationalIntegralClosure(wtr,R,I);
     -- used 0.912704 seconds

i6 : toString ic

o6 = ({f6^7,
       f11^2*f6^5-f11*f6^5,
       f11*f6^7,
       f11^5-f11^2*f6^4-3*f11^4+f11*f6^4+3*f11^3-f11^2,
       f11^3*f6^4-2*f11^2*f6^4+f11*f6^4,
       f11^4*f6^3-2*f11^3*f6^3+f11^2*f6^3},
{    p_0^2-p_0-p_2*p_5^3-p_4*p_5^5-p_4*p_5^2,
     p_0*p_1-p_0*p_5-p_3*p_5^4+p_5^4,
     p_0*p_2-p_1*p_5^3,
     p_0*p_3-p_0-p_2*p_5^3-p_4*p_5^2,
     p_0*p_4-p_1*p_5-p_4-p_5^5,
     p_1^2-p_1*p_5+p_2*p_5-p_5^5,
     p_1*p_2+p_3*p_5-p_4*p_5^3-p_5,
     p_1*p_3-p_0*p_5, p_1*p_4-p_2*p_5^2-p_4*p_5,
     p_2^2-p_0*p_5+p_1+p_3*p_5-p_5,
     p_2*p_3-p_5^4, p_2*p_4-p_3*p_5^2+p_5^2,
     p_3^2-p_3-p_4*p_5^2,
     p_3*p_4-p_1*p_5-p_4,
     p_4^2-p_0},
    QQ[p_0, p_1, p_2, p_3, p_4, p_5],
matrix{{20,  15,  13,  11,  10,   6}})
```

again piecing together results from characteristic $3, 5, 7$.

7 Timings

Here are rough timing comparisons, all done on the same departmental computer. Times vary somewhat, even on the same machine with the same problem, and certainly are machine dependent. But this clearly points out how dependent certain implementations are on the characteristic, and whether they are competitive with each other time-wise. (* means the integral closure is bigger in this char,— means not applicable, *** means time and/or space resources were exhausted.)

char	*normal*	*normalP*	*icFrac −tions*	*icFracP*	*Normal−isation*	*Integral Closure*	*M2 Qth*	*Magma Qth*
0	0.2	− − −	0.5	− − −	0.6	0.02	0.9	0.3
∗2	0.2	0.03	0.5	0.3	0.4	0.02	0.1	0.04
∗3	0.2	0.03	0.5	0.5	0.3	0.02	0.2	0.04
5	0.2	0.6	0.5	2.	0.3	0.02	0.2	0.05
7	0.2	∗ ∗ ∗	0.5	34.	0.4	0.02	0.2	0.06
11	0.2		0.5		0.3	0.01	0.4	0.09
13	0.2		0.5		0.3	0.02	0.6	0.2
17	0.2		0.5		0.3	0.01	1.	0.2
19	0.2		0.5		0.3	0.02	1.	0.2
23	0.2		0.5		0.3	0.02	2.	0.3
29	0.2		0.5		0.3	0.02	2.	0.4
31	0.2		0.5		0.3	0.01	3.	0.5
37	0.2		0.5		0.3	0.02	6.	0.6
41	0.2		0.6		0.3	0.01	7.	0.7
101	0.2		0.5		0.4	0.01	42.	4.
4001	0.2		0.5		0.3	0.02	∗ ∗ ∗	∗ ∗ ∗