

```

line = io.gets
while line = io.gets
  ar = line.chomp!.split( "\t" )
  if ar == [] then next end
  case ar[ 0 ]
  when "X"
    ar[ 0 ] = "23"
  when "Y"
    ar[ 0 ] = "24"
  when "M"
    ar[ 0 ] = "25"
  end
  ceuAll = ar[ CLCCEU ].to_f + ar[ CLCCEU + 1 ].to_f + ar[ CLCCEU + 2 ].to_f
  chbAll = ar[ CLCCHB ].to_f + ar[ CLCCHB + 1 ].to_f + ar[ CLCCHB + 2 ].to_f
  jptAll = ar[ CLCJPT ].to_f + ar[ CLCJPT + 1 ].to_f + ar[ CLCJPT + 2 ].to_f
  targetAll = ar[ Target ].to_f + ar[ Target + 1 ].to_f + ar[ Target + 2 ].to_f
  case Model
  when Ref_Dominant
    ceuDis = ar[ CLCCEU ].to_f + ar[ CLCCEU + 1 ].to_f
    chbDis = ar[ CLCCHB ].to_f + ar[ CLCCHB + 1 ].to_f
    jptDis = ar[ CLCJPT ].to_f + ar[ CLCJPT + 1 ].to_f
    targetDis = ar[ Target ].to_f + ar[ Target + 1 ].to_f
  when Ref_Recessive
    ceuDis = ar[ CLCCEU ].to_f
    chbDis = ar[ CLCCHB ].to_f
    jptDis = ar[ CLCJPT ].to_f
    targetDis = ar[ Target ].to_f
  when Alt_Dominant
    ceuDis = ar[ CLCCEU + 1 ].to_f + ar[ CLCCEU + 2 ].to_f
    chbDis = ar[ CLCCHB + 1 ].to_f + ar[ CLCCHB + 2 ].to_f
    jptDis = ar[ CLCJPT + 1 ].to_f + ar[ CLCJPT + 2 ].to_f
    targetDis = ar[ Target + 1 ].to_f + ar[ Target + 2 ].to_f
  when Alt_Recessive
    ceuDis = ar[ CLCCEU + 2 ].to_f
    chbDis = ar[ CLCCHB + 2 ].to_f
    jptDis = ar[ CLCJPT + 2 ].to_f
    targetDis = ar[ Target + 2 ].to_f
  end
  #Japanese frequency > JapaneseFrequencyThreshold
  p_jptDisFreqGTTreshold = 1.0 - math2Class.reg_incomplete_beta( JapaneseFrequencyThreshold, 1 + jptDis, 1 +
jptAll - jptDis )
  #Japanese frequency > FoldOfTargetGeneToCEU
  # p_jptDisFreqCEU = 1.0 - math2Class.reg_incomplete_beta( [ FoldOfTargetGeneToWestern * ceuDis.to_f / ceuAll.
to_f, 1.0 ].min, 1 + jptDis, 1 + jptAll - jptDis )
  p_jptDisFreqCEU = math2Class.product_reg_incomplete_beta( FoldOfTargetGeneToWestern, 1 + ceuDis, 1 + ceuAll

```

```

- ceuDis, 1 + jptDis, 1 + jptAll - jptDis )
  #Difference between CHB and JPT
  p_jptDisFreqCHB = math2Class.product_reg_incomplete_beta( FoldOfTargetGeneToChinese, 1 + chbDis, 1 + chbAll
- chbDis, 1 + jptDis, 1 + jptAll - jptDis )
# p_jptDisFreqCHB = 1.0 - math2Class.reg_incomplete_beta( [ FoldOfTargetGeneToChinese * chbDis.to_f / chbDis.
to_f, 1.0 ].min, 1 + jptDis, 1 + jptAll - jptDis )
#p p_CHB_JPT = math2Class.product_reg_incomplete_beta( 1.0, 1 + 36, 1 + 72, 1 + 36, 1 + 72 )
#p p_CHB_JPT = math2Class.product_reg_incomplete_beta( 1.0, 1 + jptDis, 1 + jptAll - jptDis, 1 + jptDis, 1 + jptAll -
jptDis )
#p [chbDis,jptDis ]###
#p_level3
p_compWorld = p_jptDisFreqGTTreshold * p_jptDisFreqCEU * p_jptDisFreqCHB
###
ioWorld.puts "#{ p_compWorld }"
ioLog.puts "#{ ar[ 0 ] }\t#{ ar[ 1 ] }\t#{ ar[ 2 ] }\t#{ ar[ 3 ] }\t#{ ar[ 4 ] }\t#{ ar[ 5 ] }\t#{ ar[ 6 ] }\t#{ ar[ 7 ] }\t#{ ar[ 8
] }\t#{ ar[ 9 ] }\t#{ ar[ 10 ] }\t#{ ar[ 11 ] }\t#{ p_jptDisFreqGTTreshold }\t#{ p_jptDisFreqCEU }\t#{ p_jptDisFreqCHB
}\t#{ p_compWorld }"
end
ioWorld.close
ioLog.close
io.close

```

C による extension

```

"math2.h"
void Init_Math2( void );
static VALUE wrap_product_reg_incomplete_beta( VALUE self, VALUE r_factor, VALUE r_a1, VALUE r_b1, VALUE r_
a2, VALUE r_b2 );
static VALUE wrap_reg_incomplete_beta( VALUE self, VALUE r_x, VALUE r_a, VALUE r_b );
static double product_reg_incomplete_beta_c( double factor, double a1, double b1, double a2, double b2 );
static double reg_incomplete_beta_c( double x, double a, double b );
static double beta( double x, double y );
static double my_gamma( double x );
static double loggamma( double x );

```

```

"math2.c"
#include "ruby.h"
#include "math.h"
#include "stdio.h"
#include <math2.h>

#define PI 3.14159265358979324 /* pi */
#define LOG_2PI 1.83787706640934548 /* log2 / pi */
#define N 8

#define BO 1 /* Bernoulli number */

```

```

#define B1 (-1.0 / 2.0)
#define B2 ( 1.0 / 6.0)
#define B4 (-1.0 / 30.0)
#define B6 ( 1.0 / 42.0)
#define B8 (-1.0 / 30.0)
#define B10 ( 5.0 / 66.0)
#define B12 (-691.0 / 2730.0)
#define B14 ( 7.0 / 6.0)
#define B16 (-3617.0 / 510.0)

static double loggamma( double x){ /* logGamma */
    double v, w;

    v = 1.0;
    while ( x < N) { v *= x; x++; }
    w = 1 / ( x * x );
    return ( ((( ((( (B16 / ( 16 * 15)) * w + (B14 / ( 14 * 13))) * w
        + (B12 / ( 12 * 11))) * w + (B10 / ( 10 * 9))) * w
        + (B8 / ( 8 * 7))) * w + (B6 / ( 6 * 5))) * w
        + (B4 / ( 4 * 3))) * w + (B2 / ( 2 * 1))) / x
        + 0.5 * LOG_2PI - log(v) - x + ( x - 0.5) * log(x);
}

static double my_gamma( double x){ /* gamma */
    if( x < 0 ) {
        return PI / ( sin(PI * x) * exp(loggamma( 1 - x )) );
    }
    return exp(loggamma( x ));
}

static double beta( double x, double y){ /* beta */
    return exp(loggamma( x ) + loggamma( y ) - loggamma( x + y ));
}

static double reg_incomplete_beta_c( double x, double a, double b){
    int k;
    double p1, q1, p2, q2, d, previous;

    if ( a <= 0 ) return HUGE_VAL;
    if ( b <= 0 ) {
        if ( x < 1 ) return 0;
        if ( x == 1 ) return 1;
        /* else */ return HUGE_VAL;
    }
    if ( x > ( a + 1 ) / ( a + b + 2 ) )

```

```

        return 1 - reg_incomplete_beta_c( 1 - x, b, a );
    if ( x <= 0 ) return 0;
    p1 = 0; q1 = 1;
    p2 = exp( a * log( x ) + b * log( 1 - x )
        + loggamma( a + b ) - loggamma( a ) - loggamma( b ) ) / a;
    q2 = 1;
    for ( k = 0; k < 200; ) {
        previous = p2;
        d = - ( a + k ) * ( a + b + k ) * x
            / ( ( a + 2 * k ) * ( a + 2 * k + 1 ) );
        p1 = p1 * d + p2; q1 = q1 * d + q2;
        k++;
        d = k * ( b - k ) * x / ( ( a + 2 * k - 1 ) * ( a + 2 * k ) );
        p2 = p2 * d + p1; q2 = q2 * d + q1;
        if ( q2 == 0 ) {
            p2 = HUGE_VAL; continue;
        }
        p1 /= q2; q1 /= q2; p2 /= q2; q2 = 1;
        if ( p2 == previous ) return p2;
    }
    printf( "regulated incomplete beta function does not converge.\n" );
    return p2;
}

static double product_reg_incomplete_beta_c( double factor, double a1, double b1, double a2, double b2 ) {
    double x = 0.0, dif = 0.01, p = 0.0;
    while( ( x * factor <= 1.0 ) && ( x + dif <= 1.0 ) ){
        p = p + ( reg_incomplete_beta_c( x + dif, a1, b1 ) - reg_incomplete_beta_c( x, a1, b1 ) ) * ( 1.0 - reg_
incomplete_beta_c( x * factor, a2, b2 ) );
        x += dif;
    }
    return p;
}

static VALUE wrap_reg_incomplete_beta( VALUE self, VALUE r_x, VALUE r_a, VALUE r_b ){
    return rb_float_new( reg_incomplete_beta_c( NUM2DBL( r_x ), NUM2DBL( r_a ), NUM2DBL( r_b ) ) );
}

static VALUE wrap_product_reg_incomplete_beta( VALUE self, VALUE r_factor, VALUE r_a1, VALUE r_b1, VALUE r_
a2, VALUE r_b2 ) {
    return rb_float_new( product_reg_incomplete_beta_c( NUM2DBL( r_factor ), NUM2DBL( r_a1 ), NUM2DBL(
r_b1 ), NUM2DBL( r_a2 ), NUM2DBL( r_b2 ) ) );
}

void Init_Math2( void ) {

```

```
volatile VALUE module;  
module = rb_define_module("Math2");  
  
rb_define_method(module, "reg_incomplete_beta", wrap_reg_incomplete_beta, 3);  
rb_define_method(module, "product_reg_incomplete_beta", wrap_product_reg_incomplete_beta, 5);  
}
```

集積症例画像集

特発性肺線維症急性増悪

掲載したものは収集症例の一部である。DAD 様の所見がある場合は、
IPF を広めに診断している

男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

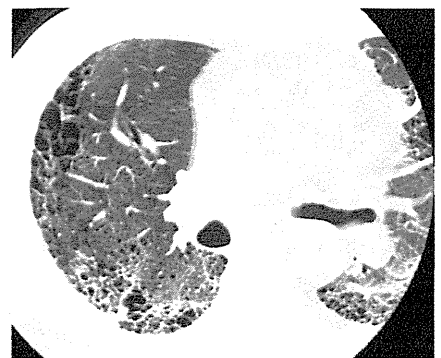
増悪後CT画像



2008/09/01



2009/04/20

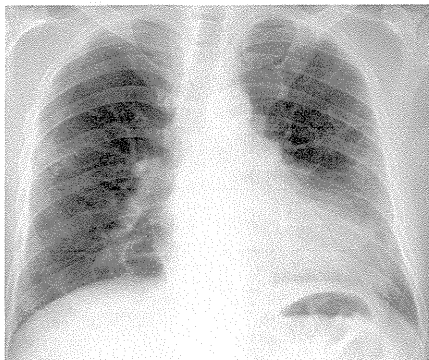


2009/05/15

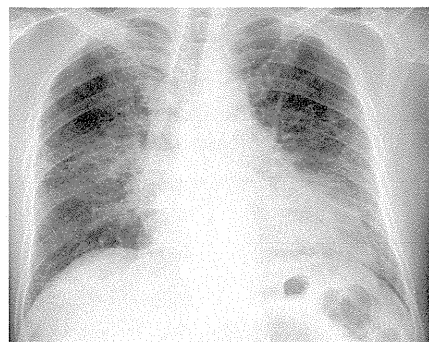
増悪前XP

増悪時XP

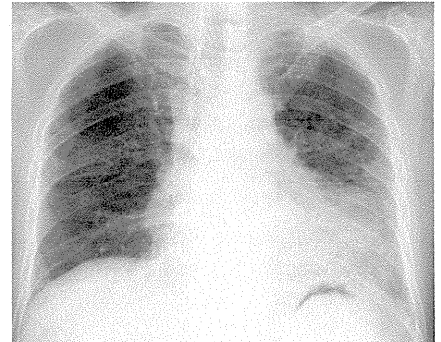
増悪後XP



2008/07/09



2009/04/20



2009/05/13

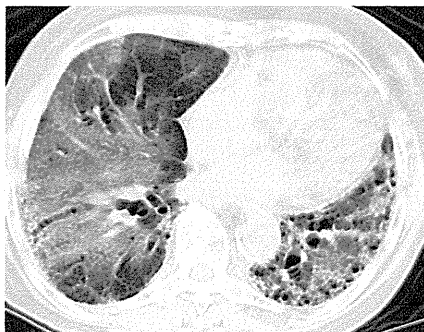
性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/02/16

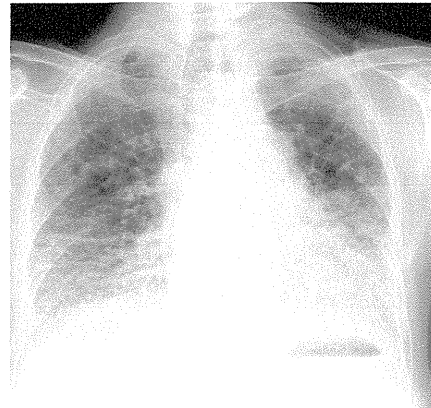
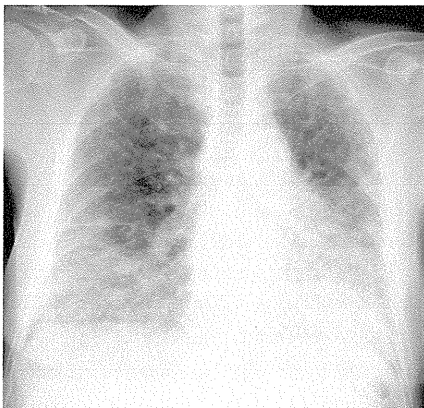
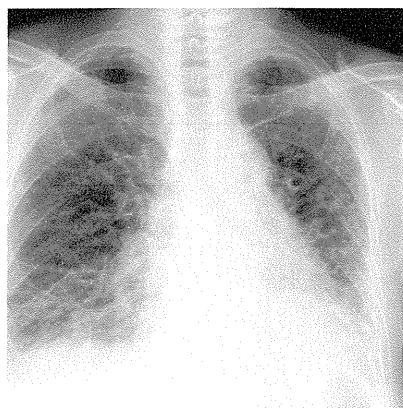
2009/04/22

2009/06/22

増悪前XP

増悪時XP

増悪後XP



2009/01/30

2009/04/22

2009/06/18

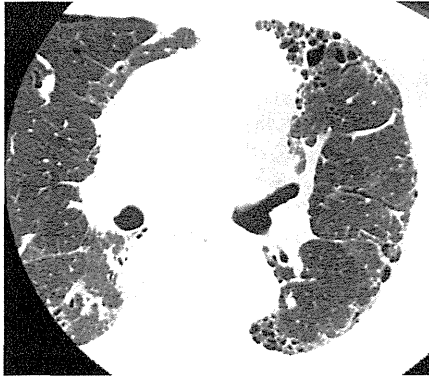
男性

IPF 急性増悪

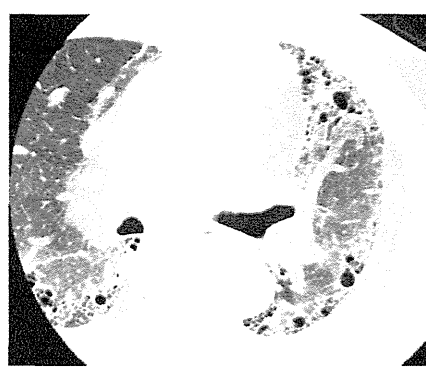
増悪前CT画像

増悪時CT画像

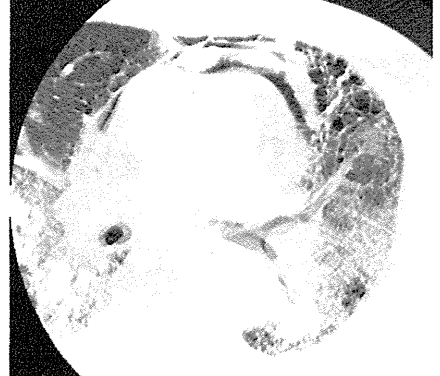
増悪後CT画像



2009/03/18



2009/05/11

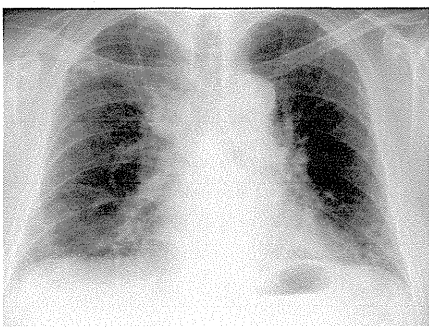


2009/06/12

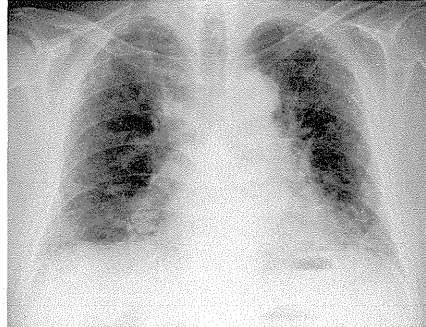
増悪前XP

増悪時XP

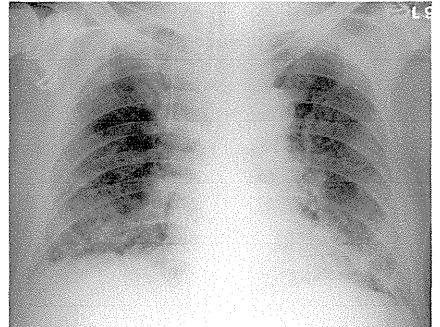
増悪後XP



2009/03/18



2009/05/11



2009/06/12

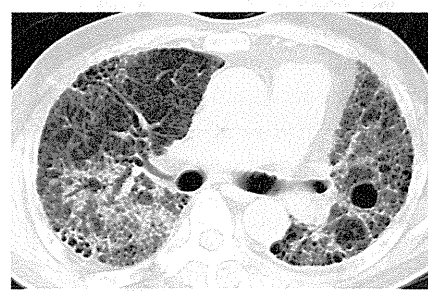
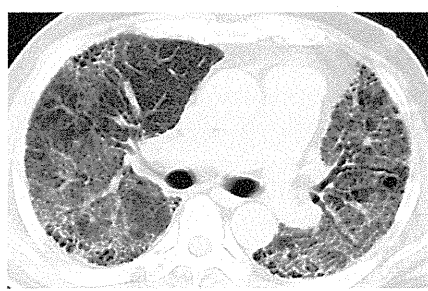
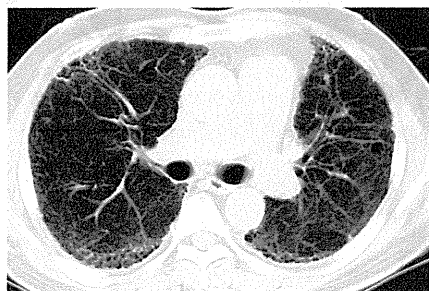
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/02/25

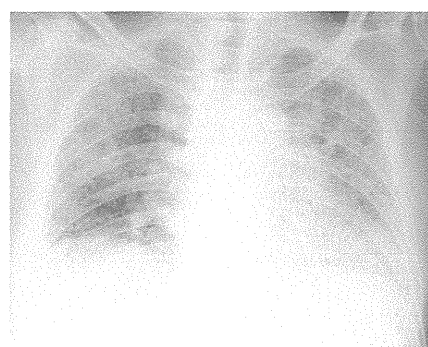
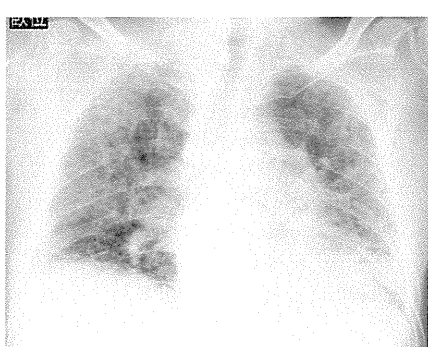
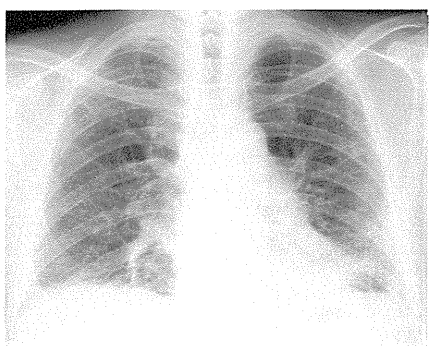
2009/06/13

2009/06/23

増悪前XP

増悪時XP

増悪後XP



2009/02/25

2009/06/13

2009/06/24

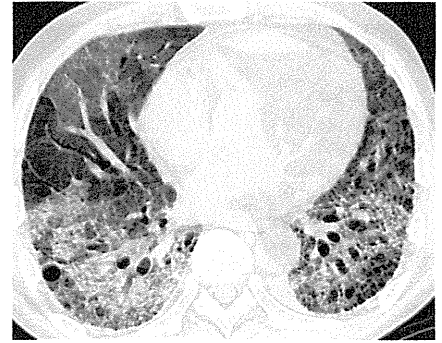
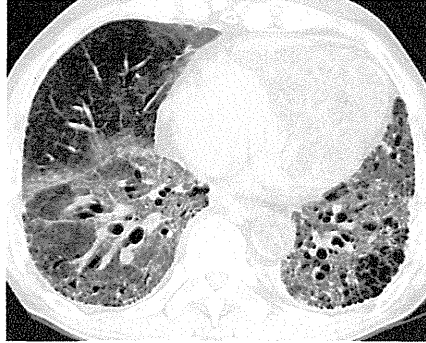
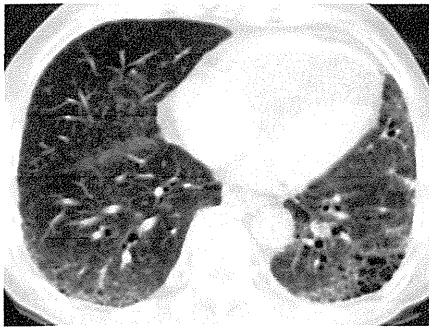
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/06/08

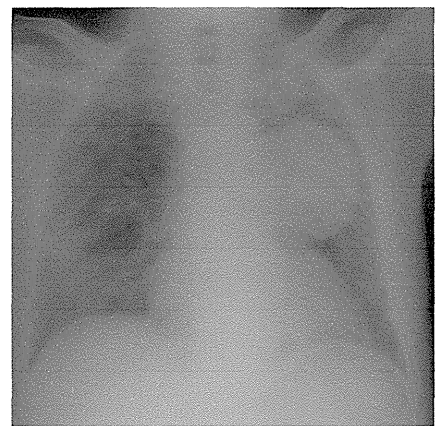
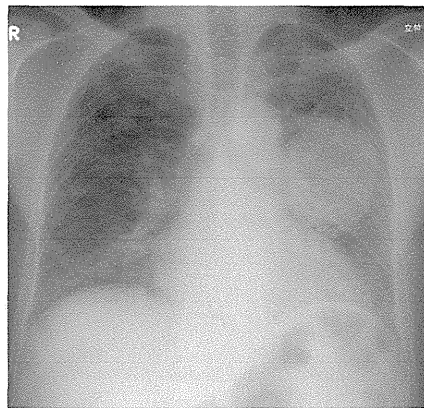
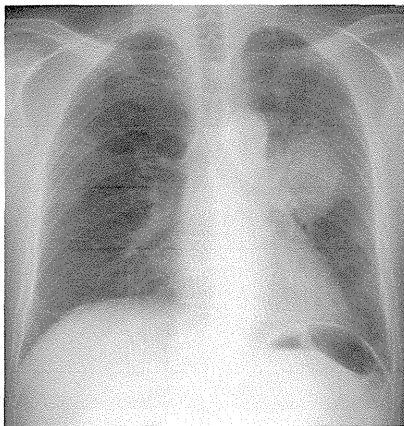
2009/06/28

2009/07/06

増悪前XP

増悪時XP

増悪後XP



2009/06/08

2009/06/28

2009/07/06

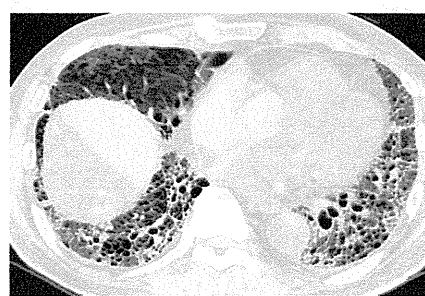
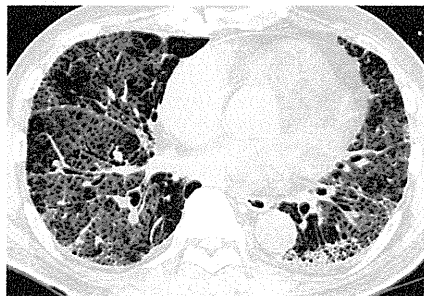
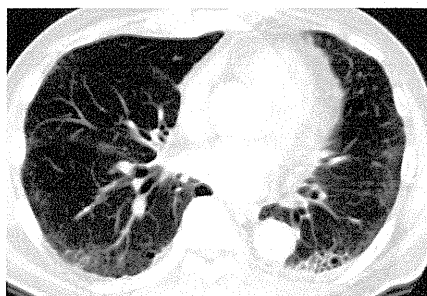
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/05/25

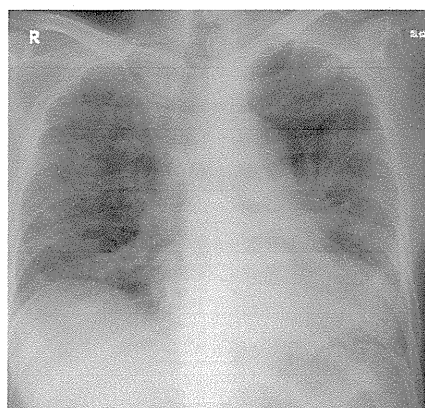
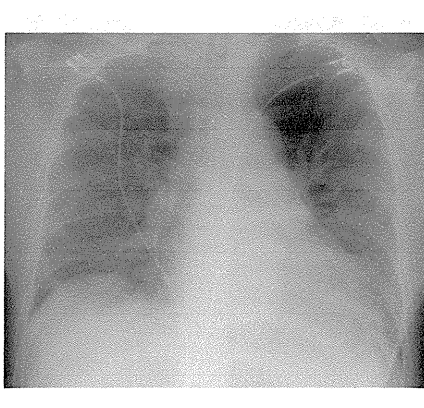
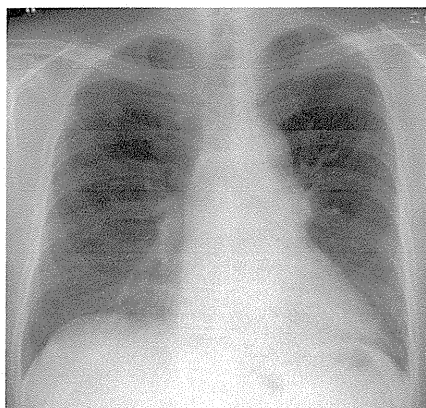
2009/07/10

2009/07/27

増悪前XP

増悪時XP

増悪後XP



2009/05/25

2009/07/10

2009/08/04

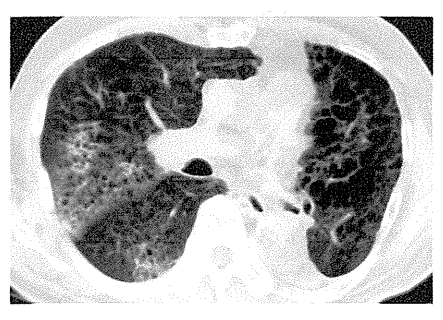
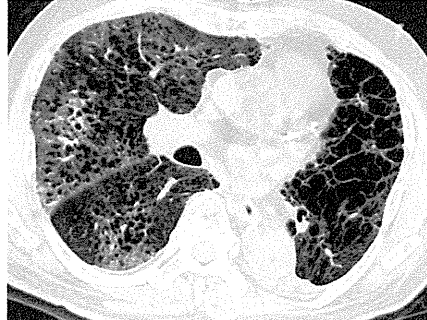
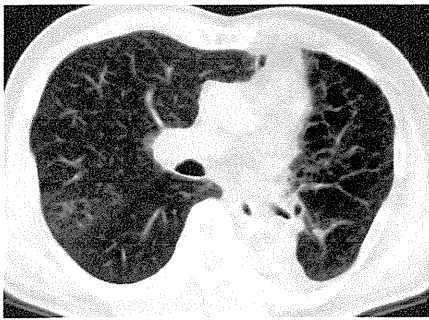
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/04/20

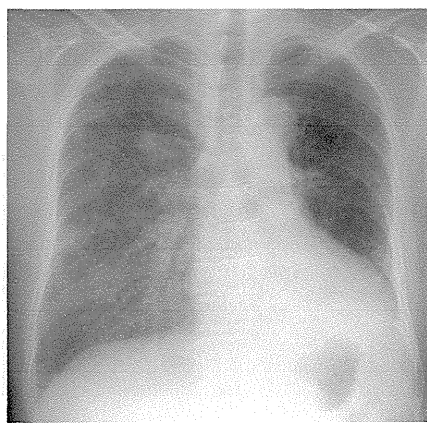
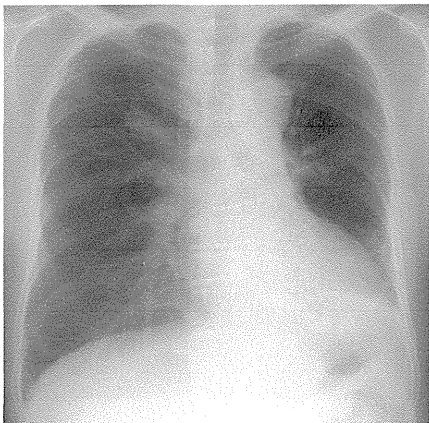
2009/07/13

2009/07/21

増悪前XP

増悪時XP

増悪後XP



2009/04/06

2009/07/13

2009/07/21

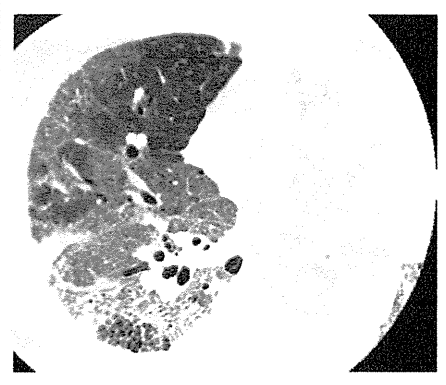
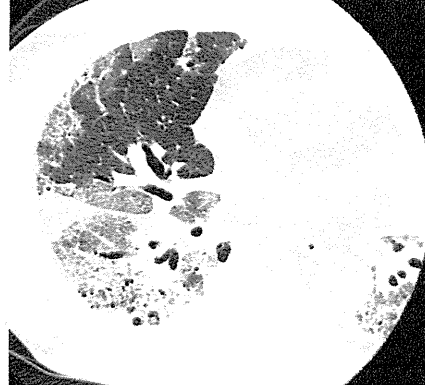
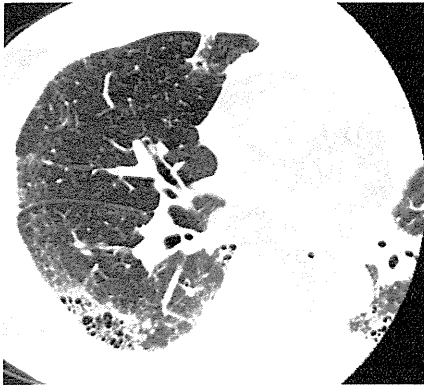
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/07/09

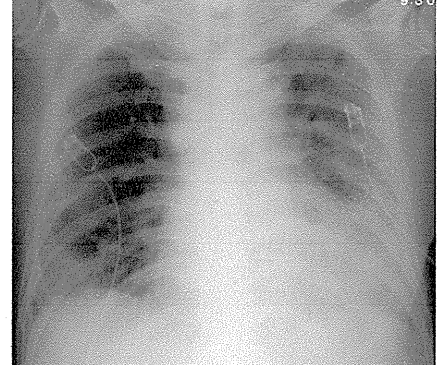
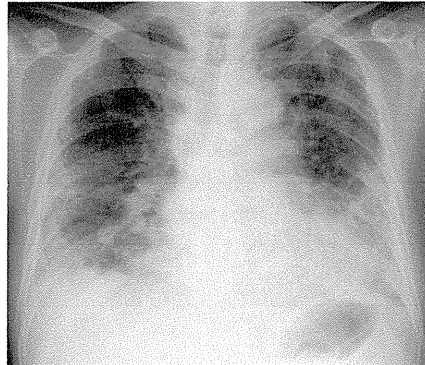
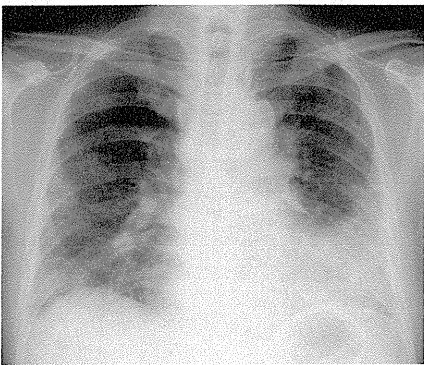
2009/08/03

2009/08/10

増悪前XP

増悪時XP

増悪後XP



2009/07/09

2009/08/03

2009/08/10

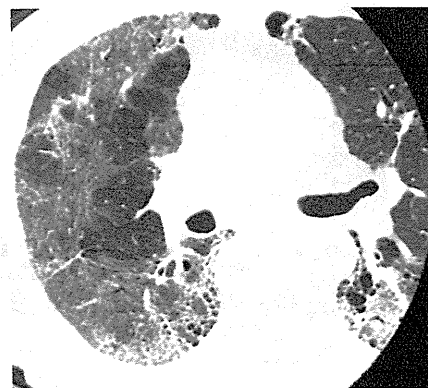
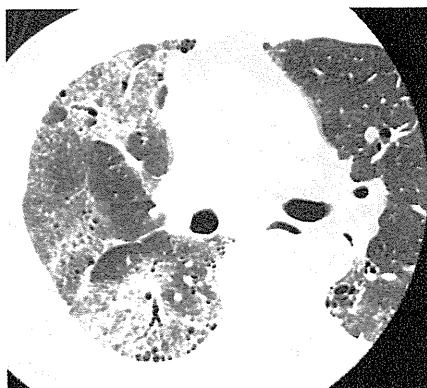
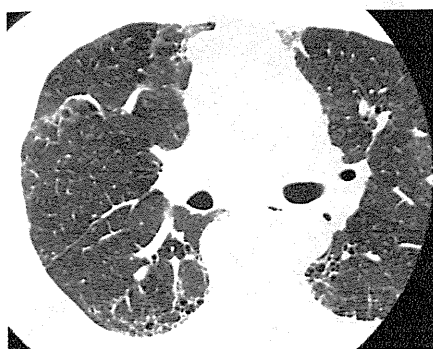
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/07/14

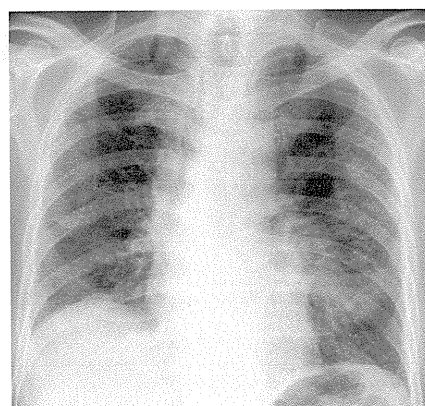
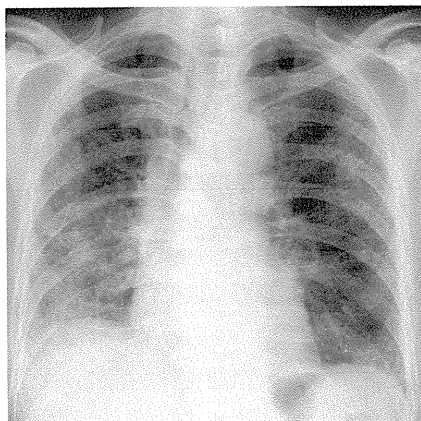
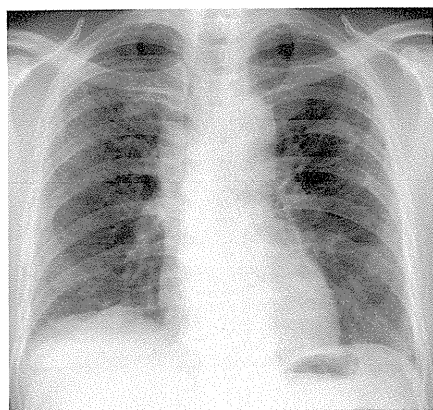
2009/08/11

2009/09/11

増悪前XP

増悪時XP

増悪後XP



2009/07/28

2009/08/11

2009/09/11

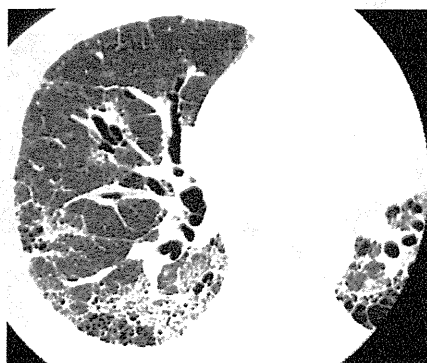
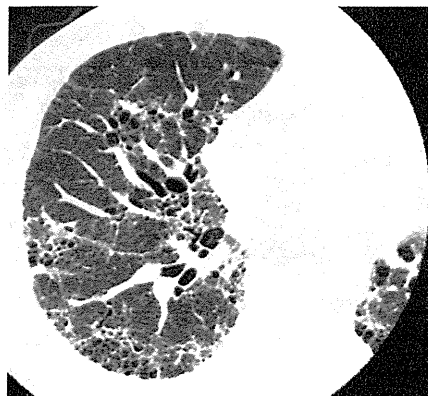
男性

IPF 急性増悪

増悪前CT画像

増悪時CT画像

増悪後CT画像



2009/08/05

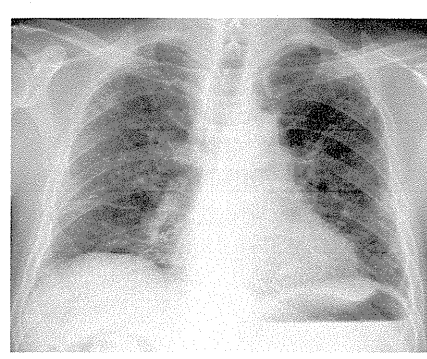
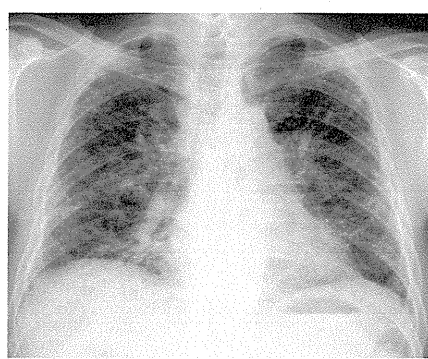
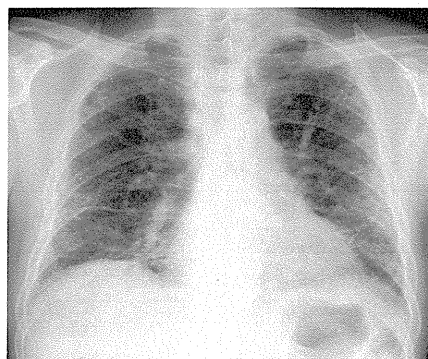
2009/08/19

2009/09/09

増悪前XP

増悪時XP

増悪後XP



2009/08/05

2009/08/19

2009/09/10

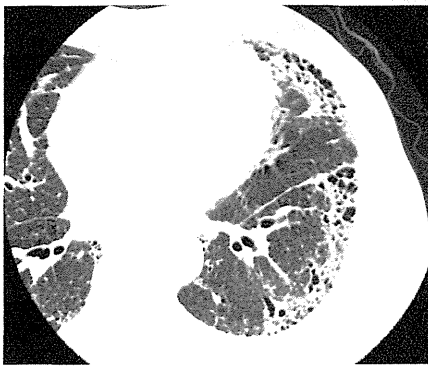
男性

IPF 急性増悪

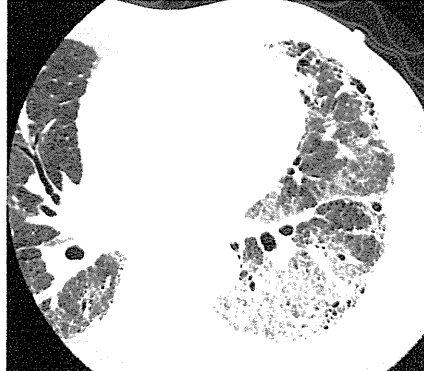
増悪前CT画像

増悪時CT画像

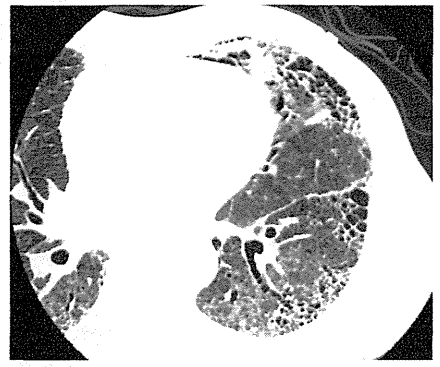
増悪後CT画像



2009/09/09



2009/12/03

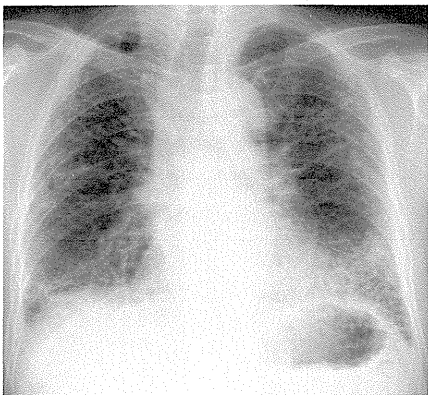


2010/01/06

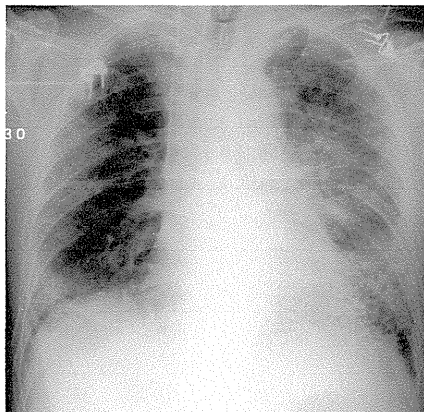
増悪前XP

増悪時XP

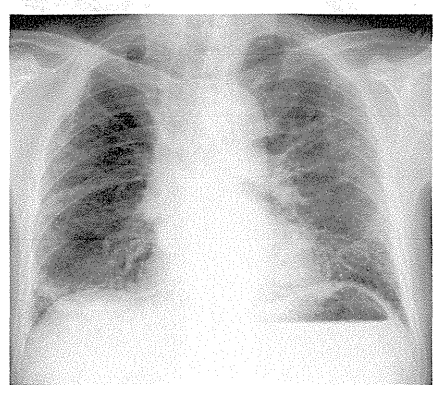
増悪後XP



2009/10/07



2009/12/03



2010/01/07

男性

IPF 急性増悪

増悪前CT画像



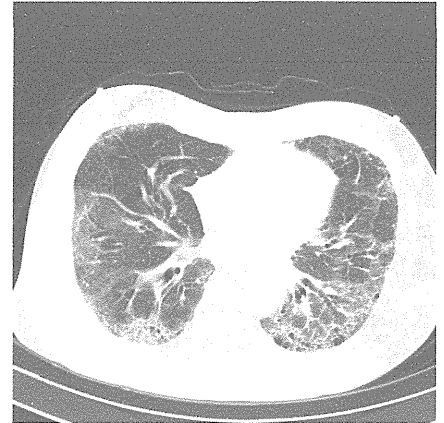
2009/08/10

増悪時CT画像



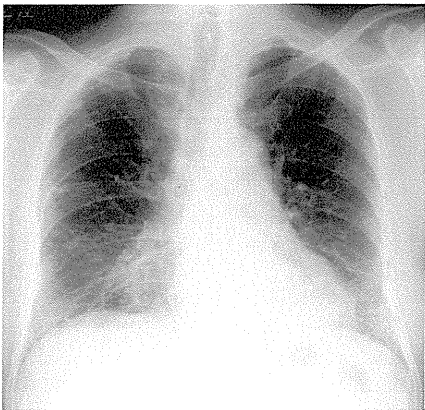
2009/12/07

増悪後CT画像



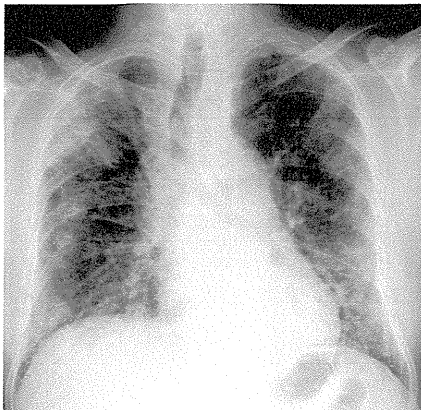
2010/01/19

増悪前XP

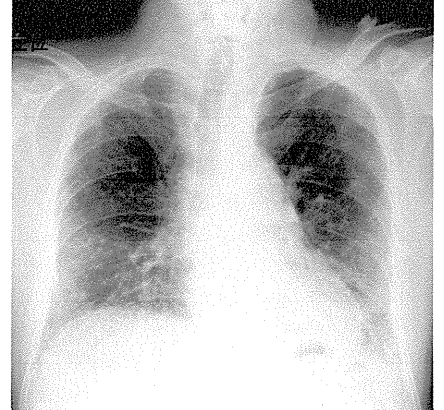


2009/08/10

増悪時XP



増悪後XP



2010/01/22

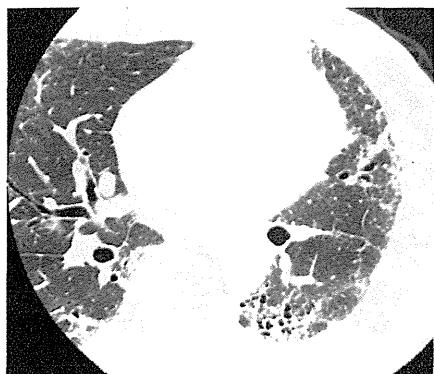
男性

IPF 急性増悪

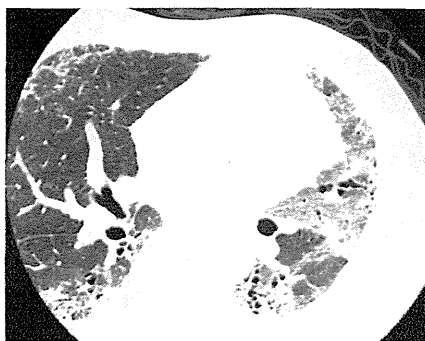
増悪前CT画像

増悪時CT画像

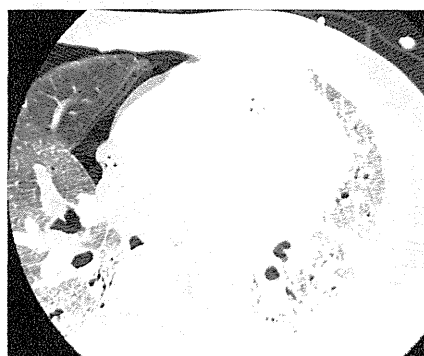
増悪後CT画像



2009/12/01



2010/01/19

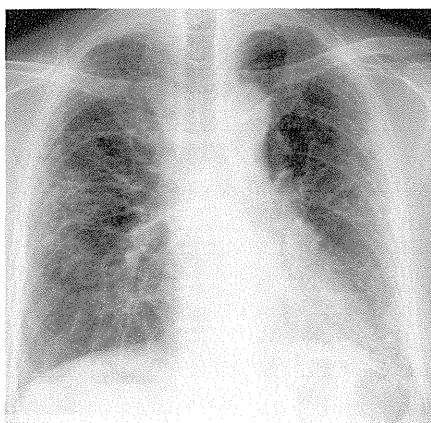


2010/01/26

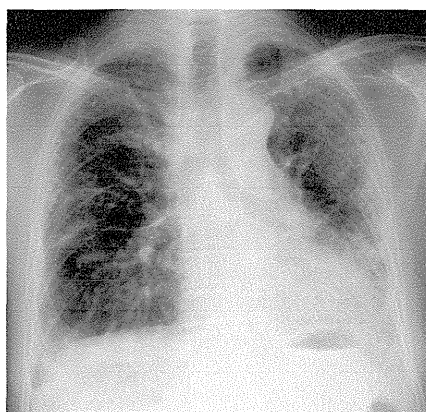
増悪前XP

増悪時XP

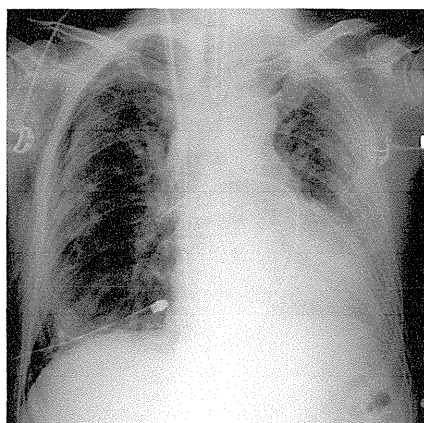
増悪後XP



2009/12/01



2010/01/19



2010/01/26