

JOSE LLOREDA CAMACHO & CO.

ABOGADOS

Calle 72 No. 5-83 Piso 5, Bogotá, D.C. Tel: 6069700 Fax 6069701

Señor

SUPERINTENDENTE DE INDUSTRIA Y COMERCIO
DIVISIÓN DE NUEVAS CREACIONES

E. _____ S. _____

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO



No. 07-104186- -00002-0000

Fecha: 2008-01-18 16:14:45 Dep. 2020 NUECREACIONES
Tra. 11 PATENTEIYII Eve: 378 FASENACIONAL
Act. 444 RESPUESTAREQUE Folios: 45

Re: P1.-GENENTECH, INC.-Solicitud de Patente de Invención Fase Nacional PCT para "COMPOSICIONES Y MÉTODOS PARA EL DIAGNÓSTICO Y TRATAMIENTO DEL TUMOR".-Clase C12N 15/52.-Expediente número 07-104.186.-

1. Me refiero al auto dictado por ese Despacho notificado por fijación en lista el 23 de Noviembre de 2007.

2. En atención al requerimiento arriba mencionado, el cuál solicita sea aclarado y traducido el contenido de los folios 160 a 190 pertenecientes al capítulo reivindicatorio, me permito manifestar a ese Despacho que la información que aparece en la Tabla 1 (folios 160 a 190) de la Solicitud de Patente en cuestión, revela el código fuente completo del programa informático ALING-2, creado por Genentech, Inc., para la comparación de secuencias de ácidos nucleicos.

Dicho código fuente está comprendido por líneas de texto escritas de acuerdo a un conjunto de reglas sintácticas propias y únicas del lenguaje de programación utilizado, que en este caso es el lenguaje de programación C.

Debido a lo anterior, la traducción al castellano de dichas líneas de texto resulta ser completamente inadecuada, toda vez que dejarían de seguir las reglas sintácticas propias y únicas del lenguaje C, perdiendo por lo tanto todo significado.

Como excepción a lo anterior, resulta de gran importancia aclarar que dentro de dicho lenguaje de programación C existe una regla de sintaxis según la cual cualquier carácter que se encuentre escrito entre los signos: /* y */, no será tenido en cuenta por el ordenador durante la fase de compilación del código fuente, por lo que dicha herramienta es utilizada frecuentemente para escribir a lo

JOSE LLOREDA CAMACHO & CO.

Calle 72 No. 5-83 Piso 5, Bogotá, D.C. Tel: 6069700 Fax 6069701

largo de dicho código palabras o frases clave que permitan realizar un seguimiento básico del significado de las variables o funciones que hacen parte del programa.

Por lo tanto, las palabras o frases que se encuentran escritas entre los signos /* y */ a lo largo del código fuente de la tabla 1, fueron traducidas literalmente al castellano.

En vista de lo anterior me permito anexar al presente, copia de los nuevos folios 160 a 190, modificados en atención a lo requerido por ese Despacho.

3. Adicionalmente, me permito presentar a ese Despacho el documento que acredita el traspaso que los autores del invento de la referencia, hacen a favor de mi representado de sus derechos sobre el mismo, debidamente apostillado y junto con su Traducción Oficial No. 122/07 debidamente legalizada hasta por el Ministerio de Relaciones Exteriores bajo el número 707707 de 2 de Enero de 2008.

4. Ruego a usted tomar atenta nota de lo anterior y proceder de conformidad.

Señor Superintendente,



ALICIA LLOREDA RICAURTE

T.P. 53.215

Tabla 1

```
/*
*
* C-C aumentado de 12 a 15
* Z es promedio de EQ
* B es promedio de ND
* coincidencia con detención es _M; detención-detención= 0; J (comodín)
coincidencia = 0 */
#define _M-8 /* valor de una coincidencia con una detención */

int _day[26][26] = {
/* A B C D E F G H I J K L M N O P Q R S T U V W X Y Z */
/* A */ { 2, 0,-2, 0, 0,-4, 1,-1,-1, 0,-1,-2,-1, 0,_M, 1, 0,-2, 1, 1, 0, 0,-6, 0,-3, 0},
/* B */ { 0, 3,-4, 3, 2,-5, 0, 1,-2, 0, 0,-3,-2, 2,_M,-1, 1, 0, 0, 0, 0,-2,-5, 0,-3, 1},
/* C */ {-2,-4,15,-5,-5,-4,-3,-3,-2, 0,-5,-6,-5,-4,_M,-3,-5,-4, 0,-2, 0,-2,-8, 0, 0,-5},
/* D */ { 0, 3,-5, 4, 3,-6, 1, 1,-2, 0, 0,-4,-3, 2,_M,-1, 2,-1, 0, 0, 0,-2,-7, 0,-4, 2},
/* E */ { 0, 2,-5, 3, 4,-5, 0, 1,-2, 0, 0,-3,-2, 1,_M,-1, 2,-1, 0, 0, 0,-2,-7, 0,-4, 3},
/* F */ {-4,-5,-4,-6,-5, 9,-5,-2, 1, 0,-5, 2, 0,-4,_M,-5,-5,-4,-3,-3, 0,-1, 0, 0, 7,-5},
/* G */ { 1, 0,-3, 1, 0,-5, 5,-2,-3, 0,-2,-4,-3, 0,_M,-1,-1,-3, 1, 0, 0,-1,-7, 0,-5, 0},
/* H */ {-1, 1,-3, 1, 1,-2,-2, 6,-2, 0, 0,-2,-2, 2,_M, 0, 3, 2,-1,-1, 0,-2,-3, 0, 0, 2},
/* I */ {-1,-2,-2,-2,-2, 1,-3,-2, 5, 0,-2, 2, 2,-2,_M,-2,-2,-2,-1, 0, 0, 4,-5, 0,-1,-2},
/* J */ { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,_M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
/* K */ {-1, 0,-5, 0, 0,-5,-2, 0,-2, 0, 5,-3, 0, 1,_M,-1, 1, 3, 0, 0, 0,-2,-3, 0,-4, 0},
/* L */ {-2,-3,-6,-4,-3, 2,-4,-2, 2, 0,-3, 6, 4,-3,_M,-3,-2,-3,-3,-1, 0, 2,-2, 0,-1,-2},
/* M */ {-1,-2,-5,-3,-2, 0,-3,-2, 2, 0, 0, 4, 6,-2,_M,-2,-1, 0,-2,-1, 0, 2,-4, 0,-2,-1},
/* N */ { 0, 2,-4, 2, 1,-4, 0, 2,-2, 0, 1,-3,-2, 2,_M,-1, 1, 0, 1, 0, 0,-2,-4, 0,-2, 1},
/* O */ { _M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,
0,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M,_M},
/* P */ { 1,-1,-3,-1,-1,-5,-1, 0,-2, 0,-1,-3,-2,-1,_M, 6, 0, 0, 1, 0, 0,-1,-6, 0,-5, 0},
/* Q */ { 0, 1,-5, 2, 2,-5,-1, 3,-2, 0, 1,-2,-1, 1,_M, 0, 4, 1,-1,-1, 0,-2,-5, 0,-4, 3},
/* R */ {-2, 0,-4,-1,-1,-4,-3, 2,-2, 0, 3,-3, 0, 0,_M, 0, 1, 6, 0,-1, 0,-2, 2, 0,-4, 0},
/* S */ { 1, 0, 0, 0, 0,-3, 1,-1,-1, 0, 0,-3,-2, 1,_M, 1,-1, 0, 2, 1, 0,-1,-2, 0,-3, 0},
/* T */ { 1, 0,-2, 0, 0,-3, 0,-1, 0, 0, 0,-1,-1, 0,_M, 0,-1,-1, 1, 3, 0, 0,-5, 0,-3, 0},
/* U */ { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,_M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
/* V */ { 0,-2,-2,-2,-2,-1,-1,-2, 4, 0,-2, 2, 2,-2,_M,-1,-2,-2,-1, 0, 0, 4,-6, 0,-2,-2},
/* W */ {-6,-5,-8,-7,-7, 0,-7,-3,-5, 0,-3,-2,-4,-4,_M,-6,-5, 2,-2,-5, 0,-6,17, 0,
0,-6},
/* X */ { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,_M, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
/* Y */ {-3,-3, 0,-4,-4, 7,-5, 0,-1, 0,-4,-1,-2,-2,_M,-5,-4,-4,-3,-3, 0,-2, 0, 0,10,-4},
/* Z */ { 0, 1,-5, 2, 3,-5, 0, 2,-2, 0, 0,-2,-1, 1,_M, 0, 3, 0, 0, 0, 0,-2,-6, 0,-4, 4}
};
```

Tabla 1 (cont')

```

/*
*/
#include <stdio.h>
#include <ctype.h>

#define MAXJMP 16 /* max saltos en una diag */
#define MAXGAP 24 /* no continuar para penalizar espacios mayores de estos */
#define JMPS 1024 /* max saltos en una ruta */
#define MX 4 /* guardar si hay al menos MX-1 bases desde el ultimo salto */
#define DMAT 3 /* valor de bases coincidentes */
#define DMIS 0 /* penalidad por bases no coincidentes */
#define DINS0 8 /* penalidad para un espacio */
#define DINS1 1 /* penalidad por base */
#define PINS0 8 /* penalidad para un espacio */
#define PINS1 4 /* penalidad por residuo */

struct jmp {
    short n[MAXJMP]; /* tamaño de salto (neg para dely) */
    unsigned short x[MAXJMP]; /* no. de base de salto en sec x */
}; /* sec limita a 2^16 -1 */

struct diag {
    int score; /* puntaje en el último salto */
    long offset; /* desviación del bloque previo */
    short ijmp; /* índice de salt. actual */
    struct jmp jp; /* lista de saltos */
};

struct path {
    int spc; /* número de espacios destacados */
    short n[JMPS]; /* tamaño de salto (espacio) */
    int x[JMPS]; /* loc del salto (último elem antes del espacio) */
};

char *ofile; /* nombre de archivo de salida */
char *namex[2]; /* nombre de sec: getseqs() */
char *prog; /* nombre del prog para err msgs */
char *seqx[2]; /* seqs: getseqs() */
int dmax; /* mejor diag: nw() */
int dmax0; /* diag final */
int dna; /* ajuste si dna: principal() */
int endgaps; /* ajuste si penalizan espacios finales */
int gapx, gapy; /* total espacios en secs */
int len0, len1; /* lons de sec */
int ngapx, ngapy; /* tamaño total de espacios */
int smax; /* puntaje max: nw() */
int *xbm; /* mapa de bit para coincidencia */

```

```
long      offset;          /* desviación actual en el archivo de salto */
struct diag *dx;          /* ocupación de diagonales */
struct path pp[2];        /* ruta de ocupación para secs */

char      *calloc(), *malloc(), *index(), *strcpy();
char      *getseq(), *g_calloc();
```

Tabla 1 (cont')

```
/* Programa de alineación de Needleman-Wunsch
*
* uso: progs archivo1 archivo2
* donde file1 y file2 son dos adn o dos secuencias de proteína.
* Las secuencias pueden estar en mayúscula o minúscula y pueden contener
ambigüedad
* cualquier línea que empieza con ';', '>' o '<' es ignorada
* longitud Max de archivo es 65535 (limitada por x corta sin firma en la estruc. del
salto)
* Se asume que la secuencia con 1/3 o más de sus elementos ACGTU es ADN
* La salida está en el archivo "align.out"
*
* El programa puede crear un archivo tmp en /tmp para retener la info acerca del
retrosegimiento.
* versión original desarrollada bajo BSD 4.3 en un vax 8650
*/
#include "nw.h"
#include "day.h"

static _dbval[26] = {
    1,14,2,13,0,0,4,11,0,0,12,0,3,15,0,0,0,5,6,8,8,7,9,0,10,0
};

static _pbval[26] = {
    1, 2|(1<<('D'-'A'))|(1<<('N'-'A')), 4, 8, 16, 32, 64,
    128, 256, 0xFFFFFFFF, 1<<10, 1<<11, 1<<12, 1<<13, 1<<14,
    1<<15, 1<<16, 1<<17, 1<<18, 1<<19, 1<<20, 1<<21, 1<<22,
    1<<23, 1<<24, 1<<25|(1<<('E'-'A'))|(1<<('Q'-'A'))
};

main(ac, av)                                main
{
    int    ac;
    char   *av[];

    prog = av[0];
    if (ac != 3) {
        fprintf(stderr,"usage: %s file1 file2\n", prog);
        fprintf(stderr,"where file1 and file2 are two dna or two protein
sequences.\n");
        fprintf(stderr,"The sequences can be in upper- or lower-case\n");
        fprintf(stderr,"Any lines beginning with ';' or '<' are ignored\n");
        fprintf(stderr,"Output is in the file \"align.out\"\n");
        exit(1);
    }
    namex[0] = av[1];
    namex[1] = av[2];
    seqx[0] = getseq(namex[0], &len0);
    seqx[1] = getseq(namex[1], &len1);
    xbm = (dna)? dbval : pbval;

    endgaps = 0;          /* 1 para penalizar espacios finales */
    ofile = "align.out";  /* archivo de salida */
}
```

```
nw();      /* llenar en la matriz, obtener posibles saltos */
readjmps(); /* obtener saltos reales */
print();   /* imprimir estat., alineación*/

cleanup(0); /* desvincular cualquier archivo tmp */}
```

Tabla 1 (cont')

/* hacer la alineación, regresar al mejor puntaje: principal()
* adn: valores en Fitch y Smith, PNAS, 80, 1382-1386, 1983
* pro: valores PAM 250
* Cuando los puntajes son iguales, preferimos no coincidencias para cualquier espacio, prefiere
* un Nuevo espacio para extender un espacio en curso, y preferir un espacio en secx
* para un espacio en sec y.
*/

nw()
{

nw

```
    char      *px, *py;          /* secs y ptrs */
    int        *ndely, *dely; /* mantener seguimiento de dely */
    int        ndelx, delx; /* mantener seguimiento de delx */
    int        *tmp;           /* para intercambio fila0, fila1 */
    int        mis;            /* puntaje para cada tipo */
    int        ins0, ins1;     /* penalidades de inserción */
    register   id;             /* índice de diagonal */
    register   ij;             /* índice de salto */
    register   *col0, *col1; /* puntaje para CURR, última fila*/
    register   xx, yy;         /* índice en secs */
```

dx = (struct diag *)g_calloc("to get diags", len0+len1+1, sizeof(struct
diag));

```
    ndely = (int *)g_calloc("to get ndely", len1+1, sizeof(int));
    dely = (int *)g_calloc("to get dely", len1+1, sizeof(int));
    col0 = (int *)g_calloc("to get col0", len1+1, sizeof(int));
    col1 = (int *)g_calloc("to get col1", len1+1, sizeof(int));
    ins0 = (dna)? DINS0 : PINS0;
    ins1 = (dna)? DINS1 : PINS1;
    smax = -10000;
    if (endgaps) {
        for (col0[0] = dely[0] = -ins0, yy = 1; yy <= len1; yy++) {
            col0[yy] = dely[yy] = col0[yy-1] - ins1;
            ndely[yy] = yy;
        }
        col0[0] = 0; /* Waterman Bull Math Biol 84 */
    }
```

else

```
        for (yy = 1; yy <= len1; yy++)
            dely[yy] = -ins0;
/* llenar en la matriz de coincidencia
*/
    for (px = seqx[0], xx = 1; xx <= len0; px++, xx++) {
        /* inicializar primera entrada en col
        */
        if (endgaps) {
            if (xx == 1)
                col1[0] = delx = -(ins0+ins1);
            else
                col1[0] = delx = col0[0] - ins1;
            ndelx = xx;
```

```
}  
else {  
    coll[0] = 0;  
    delx = -ins0;  
    ndelx = 0;  
}
```

Tabla 1 (cont')

...NW

```
DMIS;
for (py = seqx[1], yy = 1; yy <= len1; py++, yy++) {
    mis = col0[yy-1];
    if (dna)
        mis += (xbm[*px-'A']&xbm[*py-'A'])? DMAT :
    else
        mis += _day[*px-'A'][*py-'A'];

    /* actualizar penalidad para del en x sec;
    * favorecer nuevo del sobre del en curso
    * ignorar MAXGAP si son espacios finales ponderados
    */
    if (endgaps || ndely[yy] < MAXGAP) {
        if (col0[yy] - ins0 >= dely[yy]) {
            dely[yy] = col0[yy] - (ins0+ins1);
            ndely[yy] = 1;
        } else {
            dely[yy] -= ins1;
            ndely[yy]++;
        }
    } else {
        if (col0[yy] - (ins0+ins1) >= dely[yy]) {
            dely[yy] = col0[yy] - (ins0+ins1);
            ndely[yy] = 1;
        } else
            ndely[yy]++;
    }

    /* actualizar penalidad para del en sec y;
    * favorecer nuevo del sobre del en curso
    */
    if (endgaps || ndelx < MAXGAP) {
        if (col1[yy-1] - ins0 >= delx) {
            delx = col1[yy-1] - (ins0+ins1);
            ndelx = 1;
        } else {
            delx -= ins1;
            ndelx++;
        }
    } else {
        if (col1[yy-1] - (ins0+ins1) >= delx) {
            delx = col1[yy-1] - (ins0+ins1);
            ndelx = 1;
        } else
            ndelx++;
    }

    /* seleccionar el puntaje máximo; estamos favoreciendo
    * mis sobre cualquier del y delx sobre dely
```

*/

...nw

```
id = xx - yy + len1 - 1;  
if (mis >= delx && mis >= dely[yy])  
    col1[yy] = mis;
```

Tabla 1 (cont')

```

else if (delx >= dely[yy]) {
    coll[yy] = delx;
    ij = dx[id].ijmp;
    if (dx[id].jp.n[0] && (!dna || (ndelx >= MAXJMP
    && xx > dx[id].jp.x[ij]+MX) || mis >
dx[id].score+DINS0)) {
        dx[id].ijmp++;
        if (++ij >= MAXJMP) {
            writejmps(id);
            ij = dx[id].ijmp = 0;
            dx[id].offset = offset;
            offset += sizeof(struct jmp) +
sizeof(offset);
        }
        dx[id].jp.n[ij] = ndelx;
        dx[id].jp.x[ij] = xx;
        dx[id].score = delx;
    }
    else {
        coll[yy] = dely[yy];
        ij = dx[id].ijmp;
        if (dx[id].jp.n[0] && (!dna || (ndely[yy] >= MAXJMP
        && xx > dx[id].jp.x[ij]+MX) || mis >
dx[id].score+DINS0)) {
            dx[id].ijmp++;
            if (++ij >= MAXJMP) {
                writejmps(id);
                ij = dx[id].ijmp = 0;
                dx[id].offset = offset;
                offset += sizeof(struct jmp) +
sizeof(offset);
            }
            dx[id].jp.n[ij] = -ndely[yy];
            dx[id].jp.x[ij] = xx;
            dx[id].score = dely[yy];
        }
        if (xx == len0 && yy < len1) {
            /* última col
            */
            if (endgaps)
                coll[yy] -= ins0+ins1*(len1-yy);
            if (coll[yy] > smax) {
                smax = coll[yy];
                dmax = id;
            }
        }
    }
}
if (endgaps && xx < len0)
    coll[yy-1] -= ins0+ins1*(len0-xx);
if (coll[yy-1] > smax) {
    smax = coll[yy-1];
}

```

```
        dmax = id;
    }
    tmp = col0; col0 = col1; col1 = tmp;    }
(void) free((char *)ndely);
(void) free((char *)dely);
(void) free((char *)col0);
(void) free((char *)col1);    }
```

Tabla 1 (cont')

```
/*
*
* imprimir () – solamente salida visible de rutina de este módulo
*
* estático:
* getmat() – retro seguimiento de la mejor ruta, contar coincidencias: imprimir ()
* pr_align() – imprimir alineación de lo descrito en el conjunto p[]: imprimir()
* dumpblock() – descargar un bloque de líneas con números, estrellas: pr_align()
* nums() – sacar una línea de número: dumpblock()
* putline() – sacar una línea (nombre, [num], sec, [num]): dumpblock()
* stars() - - colocar una línea de estrellas: dumpblock()
* stripname() – desmontar cualquier ruta y prefijo de una seqname
*/
```

```
#include "nw.h"
```

```
#define SPC 3
```

```
#define P_LINE 256 /* línea de salida máxima */
```

```
#define P_SPC 3 /* espacio entre nombre o num y sec */
```

```
extern _day[26][26];
```

```
int olen; /* establecer longitud de la línea de salida */
```

```
FILE *fx; /* archivo de salida */
```

```
print()
```

print

```
{
    int lx, ly, firstgap, lastgap; /* solapar */

    if ((fx = fopen(ofile, "w")) == 0) {
        fprintf(stderr, "%s: can't write %s\n", prog, ofile);
        cleanup(1);
    }
    fprintf(fx, "<first sequence: %s (length = %d)\n", namex[0], len0);
    fprintf(fx, "<second sequence: %s (length = %d)\n", namex[1], len1);
    olen = 60;
    lx = len0;
    ly = len1;
    firstgap = lastgap = 0;
    if (dmax < len1 - 1) { /* espacio que guía en x */
        pp[0].spc = firstgap = len1 - dmax - 1;
        ly -= pp[0].spc;
    }
    else if (dmax > len1 - 1) { /* leading gap in y */
        pp[1].spc = firstgap = dmax - (len1 - 1);
        lx -= pp[1].spc;
    }
    if (dmax0 < len0 - 1) { /* arrastrando espacio en x */
        lastgap = len0 - dmax0 - 1;
        lx -= lastgap;
    }
}
```

```
}  
else if (dmax0 > len0 - 1) { /* trailing gap in y */  
    lastgap = dmax0 - (len0 - 1);  
    ly -= lastgap;  
}  
getmat(lx, ly, firstgap, lastgap);  
pr_align(); }
```

Tabla 1 (cont')

```

/*
 * retro seguimiento de la mejor ruta, contar coincidencias
 */
static
getmat(lx, ly, firstgap, lastgap)                                getmat
{
    int    lx, ly;                /* "núcleo" (menos espacios finales) */
    int    firstgap, lastgap;     /* conduce a seguimiento de superposición */

    int     nm, i0, i1, siz0, siz1;
    char    outx[32];
    double  pct;
    register n0, n1;
    register char *p0, *p1;
    /* obtener coincidencias totales, puntaje
     */
    i0 = i1 = siz0 = siz1 = 0;
    p0 = seqx[0] + pp[1].spc;
    p1 = seqx[1] + pp[0].spc;
    n0 = pp[1].spc + 1;
    n1 = pp[0].spc + 1;
    nm = 0;
    while ( *p0 && *p1 ) {
        if (siz0) {
            p1++;
            n1++;
            siz0--;
        }
        else if (siz1) {
            p0++;
            n0++;
            siz1--;
        }
        else {
            if (xbm[*p0-'A']&xbm[*p1-'A'])
                nm++;
            if (n0++ == pp[0].x[i0])
                siz0 = pp[0].n[i0++];
            if (n1++ == pp[1].x[i1])
                siz1 = pp[1].n[i1++];
            p0++;
            p1++;
        }
    }

    /* homología pct:
     * si se penalizan espacios finales, la base es la sec más corta
     * sino, parar salientes y tomar el núcleo más corto
     */
    if (endgaps)

```

```
        lx = (len0 < len1)? len0 : len1;
else
    lx = (lx < ly)? lx : ly;
pct = 100.*(double)nm/(double)lx;
fprintf(fx, "\n");
fprintf(fx, "<%d match%s in an overlap of %d: %.2f percent similarity\n",
        nm, (nm == 1)? "" : "es", lx, pct);
```

Tabla 1 (cont')

```

fprintf(fx, "<gaps in first sequence: %d", gapx);
if (gapx) {
    (void) sprintf(outx, " (%d %s%s)",
        ngapx, (dna)? "base":"residue", (ngapx == 1)? "" : "s");
    fprintf(fx, "%s", outx);
    fprintf(fx, ", gaps in second sequence: %d", gapy);
    if (gapy) {
        (void) sprintf(outx, " (%d %s%s)",
            ngapy, (dna)? "base":"residue", (ngapy == 1)? "" : "s");
        fprintf(fx, "%s", outx);
    }
    if (dna)
        fprintf(fx,
            "\n<score: %d (match = %d, mismatch = %d, gap penalidad = %d +
%d per base)\n",
            smax, DMAT, DMIS, DINS0, DINS1);
    else
        fprintf(fx,
            "\n<score: %d (Dayhoff PAM 250 matrix, gap penalidad = %d + %d
per residue)\n",
            smax, PINS0, PINS1);
    if (endgaps)
        fprintf(fx,
            "<endgaps penalized. left endgap: %d %s%s, right endgap: %d
%s%s\n",
            firstgap, (dna)? "base" : "residue", (firstgap == 1)? "" : "s",
            lastgap, (dna)? "base" : "residue", (lastgap == 1)? "" : "s");
    else
        fprintf(fx, "<endgaps not penalized\n");
}
static      nm;          /* coincidencias en núcleo – para revisión */
static      lmax;        /* longitudes de nombres de archivo desmontados */
static      ij[2];       /* índice de salto para una ruta */
static      nc[2];       /* número en inicio de la línea actual */
static      ni[2];       /* número de elem actual – para espacios */
static      siz[2];
static char *ps[2];      /* ptr para el elemento actual */
static char *po[2];      /* ptr para la siguiente ranura de char de salida */
static char out[2][P_LINE]; /* línea de salida */
static char star[P_LINE]; /* ajustar para estrella () */
/*
 * imprimir alineación de lo descrito en la ruta estruct pp[]
 */
static
pr_align()
{
    int      nn;          /* conteo de char */
    int      more;
    register i;

    for (i = 0, lmax = 0; i < 2; i++) {

```

...getmat

pr_align

```
nn = stripname(nameex[i]);  
if (nn > lmax)  
    lmax = nn;  
nc[i] = 1;  
ni[i] = 1;  
siz[i] = ij[i] = 0;  
ps[i] = seqx[i];  
po[i] = out[i];          }
```

Tabla 1 (cont')

```
for (nn = nm = 0, more = 1; more; ) {
    for (i = more = 0; i < 2; i++) {
        /*
         * tenemos más de esta secuencia?
         */
        if (!*ps[i])
            continue;
        more++;
        if (pp[i].spc) {          /* espacio destacado */
            *po[i]++ = ' ';
            pp[i].spc--;
        }
        else if (siz[i]) {       /* en un espacio */
            *po[i]++ = '-';
            siz[i]--;
        }
        else {                  /* estamos poniendo un elemento de sec
            */
            *po[i] = *ps[i];
            if (islower(*ps[i]))
                *ps[i] = toupper(*ps[i]);
            po[i]++;
            ps[i]++;
            /*
             * estamos en el siguiente espacio para esta secuencia?
             */
            if (ni[i] == pp[i].x[ij[i]]) {
                /*
                 * necesitamos fusionar todos los espacios
                 * en esta ubicación
                 */
                siz[i] = pp[i].n[ij[i]++];
                while (ni[i] == pp[i].x[ij[i]])
                    siz[i] += pp[i].n[ij[i]++];
            }
            ni[i]++;
        }
    }
    if (++nn == olen || !more && nn) {
        dumpblock();
        for (i = 0; i < 2; i++)
            po[i] = out[i];
        nn = 0;
    }
}
/*
 * descargar un bloque de líneas, incluyendo números, estrellas: pr_align()
```

...pr_align

```
*/  
static  
dumpblock()  
{  
    register    i;  
    for (i = 0; i < 2; i++)  
        *po[i]-- = '\0';
```

dumpblock

Tabla 1 (cont')

...dumpblock

```
(void) putc('\n', fx);
for (i = 0; i < 2; i++) {
    if (*out[i] && (*out[i] != ' ' || *(po[i]) != ' ')) {
        if (i == 0)
            nums(i);
        if (i == 0 && *out[1])
            stars();
        putline(i);
        if (i == 0 && *out[1])
            fprintf(fx, star);
        if (i == 1)
            nums(i);
    }
}
}
/*
 * sacar una línea de número: dumpblock()
 */
static
nums(ix)
    int    ix;    /* índice en salida [] línea de sec de retención */
{
    char    nline[P_LINE];
    register    i, j;
    register char *pn, *px, *py;
    for (pn = nline, i = 0; i < lmax+P_SPC; i++, pn++)
        *pn = ' ';
    for (i = nc[ix], py = out[ix]; *py; py++, pn++) {
        if (*py == ' ' || *py == '-')
            *pn = ' ';
        else {
            if (i%10 == 0 || (i == 1 && nc[ix] != 1)) {
                j = (i < 0)? -i : i;
                for (px = pn; j; j /= 10, px--)
                    *px = j%10 + '0';
                if (i < 0)
                    *px = '-';
            }
            else
                *pn = ' ';
            i++;
        }
    }
    *pn = '\0';
    nc[ix] = i;
    for (pn = nline; *pn; pn++)
        (void) putc(*pn, fx);
}
```

nums

```

        (void) putc('\n', fx);
    }
    /*
    * sacar una línea (nombre, [num], sec, [num]): dumpblock()
    */
    static
    putline(ix)                                putline
        int ix;                                {

```

Tabla 1 (cont')

...putline

```
int          i;
register char *px;

for (px = namex[ix], i = 0; *px && *px != ':'; px++, i++)
    (void) putc(*px, fx);
for (; i < lmax+P_SPC; i++)
    (void) putc(' ', fx);

/* estos cuentan desde 1:
 * ni[] es el elemento actual (desde 1)
 * nc[] es el número en inicio de la línea actual
 */
for (px = out[ix]; *px; px++)
    (void) putc(*px&0x7F, fx);
(void) putc('\n', fx);
}

/*
 * colocar una línea de estrellas (secs siempre en out[0], out[1]): dumpblock()
 */
static
stars()
{
    int          i;
    register char *p0, *p1, cx, *px;

    if (!*out[0] || (*out[0] == ' ' && *(po[0]) == ' ') ||
        !*out[1] || (*out[1] == ' ' && *(po[1]) == ' '))
        return;
    px = star;
    for (i = lmax+P_SPC; i; i--)
        *px++ = ' ';

    for (p0 = out[0], p1 = out[1]; *p0 && *p1; p0++, p1++) {
        if (isalpha(*p0) && isalpha(*p1)) {

            if (xbm[*p0-'A']&xbm[*p1-'A']) {
                cx = '*';
                nm++;
            }
            else if (!dna && _day[*p0-'A'][*p1-'A'] > 0)
                cx = '!';
            else
                cx = ' ';
        }
        else

```

stars

```
        cx = ' ';
        *px++ = cx;
    }
    *px++ = '\n';
    *px = '\0';
}
```

Tabla 1 (cont')

```
/*
 * desmontar ruta o prefijo de pn, volver a len: pr_align()
 */
static
stripname(pn)
    char *pn; /* nombre de archivo (puede ser la ruta) */
{
    register char *px, *py;

    py = 0;
    for (px = pn; *px; px++)
        if (*px == '/')
            py = px + 1;
    if (py)
        (void) strcpy(pn, py);
    return(strlen(pn));
}
```

stripname

Tabla 1 (cont')

```

/*
* cleanup() – limpiar cualquier archivo tmp
* getseq() – leer en la sec, establecer adn, len, maxlen
* g_calloc() -- calloc() con verificación, de error
* readjumps() – obtener los buenos saltos, del archivo tmp si es necesario
* writejumps() – escribir un conjunto llenado de saltos en un archivo tmp: nw()
*/
#include "nw.h"
#include <sys/file.h>

char *jname = "/tmp/homgXXXXXX";          /* archivo tmp para saltos */
FILE *fj;
int cleanup();                          /* limpiar archivo tmp */
long lseek();
/*
* remover cualquier archivo tmp si nos pasamos
*/
cleanup(i)                                cleanup
{
    int i;
    if (fj)
        (void) unlink(jname);
    exit(i);
}
/*
* leer, volver ptr a sec, ajustar adn, len, maxlen
* desmontar líneas que empiezan con ';', '<', o '>'
* sec en mayúscula o minúscula
*/
char *
getseq(file, len)                        getseq
{
    char line[1024], *pseq;
    register char *px, *py;
    int natgc, tlen;
    FILE *fp;
    if ((fp = fopen(file, "r")) == 0) {
        fprintf(stderr, "%s: can't read %s\n", prog, file);
        exit(1);
    }
    tlen = natgc = 0;
    while (fgets(line, 1024, fp)) {
        if (*line == ';' || *line == '<' || *line == '>')
            continue;
        for (px = line; *px != '\n'; px++)
            if (isupper(*px) || islower(*px))

```

```
        tlen++;
    }
    if ((pseq = malloc((unsigned)(tlen+6))) == 0) {
        fprintf(stderr, "%s: malloc() failed to get %d bytes for %s\n", prog,
tlen+6, file);
        exit(1);
    }
    pseq[0] = pseq[1] = pseq[2] = pseq[3] = '\0';
```

331

Tabla 1 (cont')

...getseq

```

py = pseq + 4;
*len = tlen;
rewind(fp);
while (fgets(line, 1024, fp)) {
    if (*line == ';' || *line == '<' || *line == '>')
        continue;
    for (px = line; *px != '\n'; px++) {
        if (isupper(*px))
            *py++ = *px;
        else if (islower(*px))
            *py++ = toupper(*px);
        if (index("ATGCU", *(py-1)))
            natgc++;
    }
}
*py++ = '\0';
*py = '\0';
(void) fclose(fp);
dna = natgc > (tlen/3);
return(pseq+4);
}
char *
g_calloc(msg, nx, sz)
char *msg; /* programa, llamando rutina */
int nx, sz; /* número y tamaño de elementos */
{
    char *px, *calloc();
    if ((px = calloc((unsigned)nx, (unsigned)sz)) == 0) {
        if (*msg) {
            fprintf(stderr, "%s: g_calloc() failed %s (n=%d, sz=%d)\n",
prog, msg, nx, sz);
            exit(1);
        }
    }
    return(px);
}

/*
* obtener saltos finales de dx[] o archivo tmp, establecer pp[], reajustar dmax: main()
*/
read_jmps ()
{
    int fd = -1;
    int siz, i0, i1;
    register i, j, xx;
    if (fj) {
        (void) fclose(fj);

```

g_calloc

read_jmps

```
    if ((fd = open(jname, O_RDONLY, 0)) < 0) {  
        fprintf(stderr, "%s: can't open() %s\n", prog, jname);  
        cleanup(1);  
    }  
}  
for (i = i0 = i1 = 0, dmax0 = dmax, xx = len0; ; i++) {  
    while (1) {  
        for (j = dx[dmax].ijmp; j >= 0 && dx[dmax].jp.x[j] >= xx; j--)  
            ;  
    }
```

933

Tabla 1 (cont')

...readjumps

```

    if (j < 0 && dx[dmax].offset && fj) {
        (void) lseek(fd, dx[dmax].offset, 0);
        (void) read(fd, (char *)&dx[dmax].jp, sizeof(struct
jmp));
        (void) read(fd, (char *)&dx[dmax].offset,
sizeof(dx[dmax].offset));
        dx[dmax].ijmp = MAXJMP-1;
    }
    else
        break;
    if (i >= JMPS) {
        fprintf(stderr, "%s: too many gaps in alignment\n", prog);
        cleanup(1);
    }
    if (j >= 0) {
        siz = dx[dmax].jp.n[j];
        xx = dx[dmax].jp.x[j];
        dmax += siz;
        if (siz < 0) { /* espacio en segunda seq */
            pp[1].n[i1] = -siz;
            xx += siz;
            /* id = xx - yy + len1 - 1 */
            pp[1].x[i1] = xx - dmax + len1 - 1;
            gapy++;
            ngapy -= siz;
            /* ignorar MAXGAP cuando se hacen espacios finales */
            siz = (-siz < MAXGAP || endgaps)? -siz : MAXGAP;
            i1++;
        }
        else if (siz > 0) { /* gap in first seq */
            pp[0].n[i0] = siz;
            pp[0].x[i0] = xx;
            gapx++;
            ngapx += siz;
            /* ignorar MAXGAP cuando se hacen espacios finales */
            siz = (siz < MAXGAP || endgaps)? siz : MAXGAP;
            i0++;
        }
    }
    else
        break;
}
/* revocar la orden de saltos */
for (j = 0, i0--; j < i0; j++, i0--) {
    i = pp[0].n[j]; pp[0].n[j] = pp[0].n[i0]; pp[0].n[i0] = i;
    i = pp[0].x[j]; pp[0].x[j] = pp[0].x[i0]; pp[0].x[i0] = i;
}
for (j = 0, i1--; j < i1; j++, i1--) {
    i = pp[1].n[j]; pp[1].n[j] = pp[1].n[i1]; pp[1].n[i1] = i;
    i = pp[1].x[j]; pp[1].x[j] = pp[1].x[i1]; pp[1].x[i1] = i;
}

```

```
if (fd >= 0)
    (void) close(fd);
if (fj) {
    (void) unlink(jname);
    fj = 0;
    offset = 0;
}
```

335

Tabla 1 (cont')

```

/*
 * escribir una desviación de struct. de salto llenada de la prev (si la hay): nw()
 */
writejumps(ix)                                     writejumps
{
    int ix;

    char *mktemp();

    if (!fj) {
        if (mktemp(jname) < 0) {
            fprintf(stderr, "%s: can't mktemp() %s\n", prog, jname);
            cleanup(1);
        }
        if ((fj = fopen(jname, "w")) == 0) {
            fprintf(stderr, "%s: can't write %s\n", prog, jname);
            exit(1);
        }
    }
    (void) fwrite((char *)&dx[ix].jp, sizeof(struct jmp), 1, fj);
    (void) fwrite((char *)&dx[ix].offset, sizeof(dx[ix].offset), 1, fj);
}

```

336

TRADUCCIÓN OFICIAL No. 122/07 de un documento escrito en inglés, el cual para su identificación se sella con el sello del Traductor. Esta es una Traducción Oficial efectuada el día 28 de diciembre de 2007 por el Sr. Gabriel Ricaurte Echeverri, c. c. 25965 de Bogotá, Traductor Oficial según resolución No. 3987 de agosto 17 de 1981 expedida por el Ministerio de Justicia.

= = =

Apostilla

(Convención de la Haya del 5 de octubre de 1961)

1. País: Estados Unidos de América

Este documento público

2. ha sido firmado por **M. Tarver**

3. Actuando en su condición de **Funcionario Certificador**

4. Lleva el sello/estampilla de la Oficina de Patentes y Marcas

Certificado

5. Washington D.C.
6. Doce de diciembre de 2007
7. Por el Funcionario Asistente de Autenticaciones, Departamento de Estado de Estados Unidos
8. No. 08007053-1

9. Sello/Estampilla: Sello en alto relieve del Departamento de Estado de Estados Unidos de América con escudo en el centro.	10. Firma (Fdo.) Sonya M. Johnson
--	--

= = =

ESTADOS UNIDOS DE AMÉRICA

A TODOS A QUIENES LLEGUE EL PRESENTE DOCUMENTO:

GABRIEL RICAURTE ECHEVERRI
TRADUCTOR OFICIAL
RES. 3987 Ministerio de Justicia
de agosto 17 de 1981

337

DEPARTAMENTO DE COMERCIO DE LOS ESTADOS UNIDOS
OFICINA DE PATENTES Y MARCAS DE LOS ESTADOS UNIDOS

Noviembre 20 de 2007

ESTO ES PARA CERTIFICAR QUE LA AQUI ADJUNTA ES UNA COPIA FIEL DE
LOS REGISTROS DE ESTA OFICINA DE UN DOCUMENTO INSCRITO EL 8 DE
AGOSTO DE 2005

Por Autoridad de
Sub-secretario de Comercio para Propiedad Intelectual y Director de la
Oficina de Patentes y Marcas de Estados Unidos

(Firmado) M. TARVER

Funcionario Certificador

Sello en papel metalizado en alto relieve del Departamento de Comercio, Oficina de
Patentes y Marcas con escudo en el centro.

Agosto 8 de 2005 11:05 AM de GENENTECH LEGAL T-682 P002/004 F-864

HOJA DE PORTADA DE FORMULARIO DE INSCRIPCIÓN
PATENTES ÚNICAMENTE

Departamento de Comercio de EU
Oficina de Patentes y Marcas
Atn. Exp. P5001R1P2

Comisionado Asistente de Patentes: Favor inscribir los documentos originales
adjuntos o copia de los mismos.

1. Nombre de la parte Cedente:

Patrick Dowd, Gretchen Frantz, Paul Polakis, Victoria Smith, Susan D. Spenser,
Thomas D. Wu, Zemin Zhang se adjuntan nombres adicionales de la parte
cedente Si x No

2. Nombre y dirección de la parte receptora:

Nombre: GENENTECH INC.

Dirección Internacional:

GABRIEL RICALVATE ECHEVERRI
TRANSCRIBIDOR OFICIAL
RES. 8837 Ministerio de Justicia
de agosto 17 de 1981

338

Dirección: <u>1 DNA Way</u> Ciudad: <u>San Francisco del Sur</u> Estado: <u>California</u> Código Postal: <u>94080-4990</u> se adjuntan nombres y direcciones adicionales: <u>Si x</u> <u>No</u> 3. Naturaleza de la cesión: <u>X</u> Traspaso Fecha de Otorgamiento: <u>7/14/05, 7/18/05, 7/15/05, 7/15/05, 7/15/05, 7/15/05, y 7/15/05, respectivamente</u>	
4. Número de solicitud o número de la patente Si este documento está siendo presentado junto con una nueva solicitud, la fecha de otorgamiento de la solicitud es: a) Solicitud de patente Nro.: <u>11/102,501</u> b) Patente Nro.: _____ Se adjuntan números adicionales <u>Si x</u> <u>No</u>	
5. Nombre y dirección de la parte a quien se debe dirigir la correspondencia con respecto al documento: Mark T. Kresnak, Ph.D GENENTECH INC. 1 DNA Way San Francisco del Sur, CA 94080 6. Número total de solicitudes y patentes involucradas: <u>1</u> 7. Total derechos (37 CFR 3.41) \$40,00 Adjunto <u>X</u> autorizados para ser cargados a la cuenta de depósito 8. Cuenta de depósito número: <u>07-0630</u>	
No use este espacio	
9. Declaración y firma Para lo mejor de mi conocimiento y creencia, la información anterior es verdadera y correcta y cualquier copia adjunta es una copia fiel del documento original.	

GABRIEL RICAURTE ECHEVERRI
TRADUCTOR OFICIAL
RES. 3967 Ministerio de Justicia
de agosto 17 de 1981

339

<u>Mark T. Kresnak, Ph.D</u>	<u>Firmado ilegiblemente</u>	<u>8/8/05</u>
Nombre de la persona firmante	Firma	fecha
Registro Nro. 42767		
Número total de páginas que comprenden la hoja de portada, anexos y documentos: <u>4</u>		
PATENTE		
CARRETE: 016618 0262		

700201259

===

Agosto-08-2005 11:06 am De GENENTECH LEGAL T-682 P003/004 F-864

Inventores Conjuntos

Expediente No. P5001R1P2

CESION

CONSIDERANDO,

Patrick Dowd, un ciudadano de los EU, residente en 18 Ord Street, San Francisco, CA 94114, y Gretchen Frantz, un ciudadano de los EU, residente en 2819 Buchanan Street, San Francisco, CA 94123, y Paul Polakis, un ciudadano de los EU, residente en 1449 Cortez Ave, Burlingame, CA 94010, y Victoria Smith, una ciudadana de AU, residente en 19 Dwight Road, Burlingame, CA 94010, y Susan D. Spenser, una ciudadana de EU, residente en 2395 Paradise Drive, Tiburón CA 49920 y Thomas D. Wu, un ciudadano de los EU, residente en 41 Nevada Street, San Francisco, CA 94110, y Zemin Zhang, un ciudadano de los EU, residente en 876 Taurus Drive, Foster City, CA 94904, (en adelante "CEDENTES"), han inventado una nueva y útil invención en

Composiciones y métodos para el Diagnóstico y Tratamiento del Tumor

Por la cual ha sido presentado por ellos el 8 de abril de 2005 una solicitud con número de serie 11/102,501 (Expediente Nro. P5001R1P2) para Privilegio de Patente de Invención; y

CONSIDERANDO QUE, GENENTECH INC., una sociedad organizada y existente bajo y por virtud de las Leyes del Estado de Delaware, teniendo como

GABRIEL RICARDO ECHEVERRI
TRADUCTOR OFICIAL
RES. 3987 Ministerio de Justicia
de agosto 17 de 1981

340

lugar de negocios en 1 DNA Way, San Francisco del Sur, California 9408-4990, está deseosa de adquirir un interés en y para la mencionada invención, y en y para el Privilegio de Patente de Invención para ser obtenido para la misma;

AHORA, POR LO TANTO, siendo conocido por todos a quienes puede interesar;

Que por una contraprestación buena y valiosa el recibo de la cual por este acto es reconocido, los mencionados cedentes por el presente acto venden, ceden, transfieren y legalmente establecido al mencionado GENENTECH INC. sus sucesores y cesionarios el derecho total y exclusivo, título e interés, incluyendo todos los derechos bajo la Convención de París para la protección de la Propiedad Industrial en y para la mencionada invención, en y para cualquiera y todos los privilegios de patente de invención que se concederán y expedirán para los mismos o cualquier continuación, división, renovación o sustituto de la misma y respecto de los privilegios de patente de invención cualquier reexpedición o reevaluación de las mismas, no solo para y en los Estados Unidos de América, sus territorios y posesiones sino también para cualquier otro país; y se ha autorizado y solicitado por medio de este acto que las agencias gubernamentales apropiadas expidan el mencionado privilegio de patente de invención a la mencionada GENENTECH INC. de acuerdo con este cesión.

Los mencionados cedentes pactan y acuerdan cooperar con GENENTECH INC. para habilitar a la mencionada GENENTECH INC. para disfrutar en la máxima extensión el derecho título e interés aquí mismo cedido en los Estados Unidos y en los países extranjeros. Tal cooperación por parte de los mencionados cedentes incluye la producción pronta de hechos y documentos pertinentes, rindiendo testimonio otorgamiento de peticiones, juramentos, especificaciones, declaraciones u otros documentos, y otra asistencia toda en la medida que se considere necesaria o deseable por parte de la mencionada GENENTECH INC; (a) para perfeccionar el derecho, título e interés aquí mismo cedido; (b) para el desarrollo de cualquiera de las mencionadas solicitudes; (c) para presentar y desarrollar solicitudes para reexpedición de cualquiera de las mencionadas patentes; (d) para la interferencia u otro procedimiento prioritario que involucre la mencionada invención; y (e) por los

GABRIEL RICARDO ECHAVEZ
TRADUCTOR OFICIAL
RES. 3987 Ministerio de Justicia
de agosto 17 de 1981

procedimientos legales que involucran la mencionada invención y cualquiera de las solicitudes de la misma y cualquiera de las patentes concedidas para la misma, incluyendo sin limitación procedimientos de oposición, cancelación, recursos de prioridad, procedimientos de uso público, acciones por infracción y acciones ante cortes; siempre que los gastos incurridos por parte de los mencionados cedentes en proveer tal cooperación sean pagados por la menciona GENENTECH INC.

PATENTE

Carrete: 016618 MARCO 0263

Los términos y pactos de esta cesión tendrán efectos para el beneficio de la mencionada GENENTECH INC., sus sucesores, cesionarios y otros representantes legales y será vinculante frente a los mencionados cedentes sus respectivos herederos, representantes legales y sus cesionarios.

Los mencionados cedentes por este acto garantizan y afirman que ellos no han llegado y no harán ninguna cesión, contrato o entendimiento con el presente.

EN TESTIMONIO DE LO CUAL nosotros firmamos como sigue:

Fecha: 7/14/05	Patrick Dowd,
Fecha: 7/18/05	Gretchen Frantz,
Fecha: 7/15/05	Paul Polakis,
Fecha: 7/15/05	Victoria Smith,
Fecha: 7/15/05	Susan D. Spenser,
Fecha: 7/15/05	Thomas D. Wu,
Fecha: 7/15/05	Zemin Zhang

San Francisco del Sur

INSCRITO EL 08/08/2005

PATENTE

CARRETE: 016618 MARCO: 0264

===

Es traducción fiel y completa de un documento escrito en inglés del cual queda copia en esta oficina ya la cual me remito para su confrontación.

Derechos de traducción: \$ 102.000,00

GABRIEL RICAURTE ECHEVERRI
TRADUCTOR OFICIAL
RES. 3987 Ministerio de Justicia
de agosto 17 de 1981

342

APOSTILLE

(Convention de La Haye du 5 octobre 1961)

1. Country: *United States of America*

This public document

2. has been signed by M. Tarver

3. acting in the capacity of Certifying Officer

4. bears the seal/stamp of Patent and Trademark Office

Certified

5. at Washington, D.C.

6. the twelfth of December, 2007

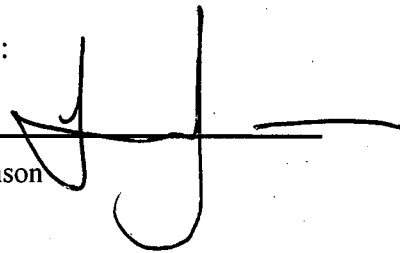
7. by *Assistant Authentication Officer, United States Department of State*

8. No. 08007053-1

9. Seal/Stamp:

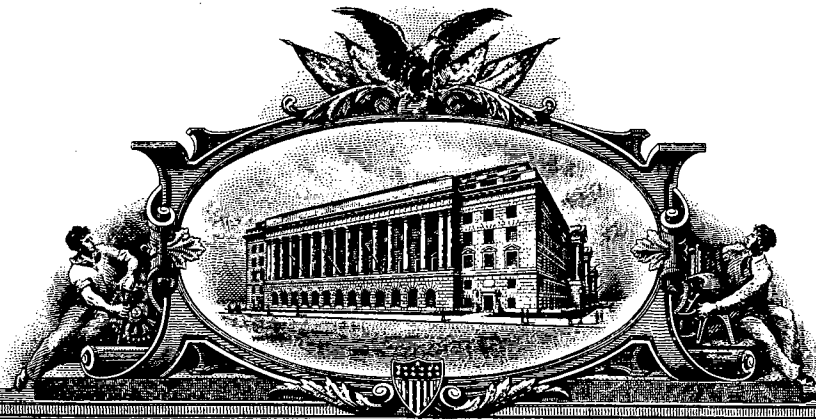
10. Signature:

Sonya N. Johnson



Columbia 343

A 7098540



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

November 20, 2007

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE
RECORDS OF THIS OFFICE OF A DOCUMENT RECORDED ON
August 8, 2005.

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office



M. Tarver

M. TARVER
Certifying Officer

344

Aug-08-2005 11:05am From-GENENTECH LEGAL

T-682 P.002/004 F-864

RECORDATION FORM COVER SHEET PATENTS ONLY		U.S. DEPARTMENT OF COMMERCE Patent and Trademark Office Any Dkt No: P5001R1P2
Assistant Commissioner of Patents: Please record the attached original documents or copy thereof.		
1. Name of conveying party(ies): Patrick Dowd, Gretchen Frantz, Paul Polakis, Victoria Smith, Susan D. Spencer, Thomas D. Wu, Zemin Zhang Additional name(s) of conveying party(ies) attached? ☐ Yes ☒ No	2. Name and address of receiving party(ies): Name: <u>GENENTECH, INC.</u> Internal Address: Street Address: <u>1 DNA Way</u> City: <u>South San Francisco</u> State: <u>California</u> ZIP: <u>94080-4990</u> Additional name(s) & address(es) attached? ☐ Yes ☒ No	
3. Nature of conveyance: ☒ Assignment	Execution Date: <u>7/14/05, 7/18/05, 7/15/05, 7/15/05, 7/15/05,</u> <u>7/15/05, and 7/15/05, respectively.</u>	
4. Application number(s) or patent number(s): If this document is being filed together with a new application, the execution date of the application is: A. Patent Application No.(s) 11/102,501 B. Patent No.(s) Additional numbers attached? ☐ Yes ☒ No		
5. Name and address of party to whom correspondence concerning document should be mailed: Mark T. Kresnak, Ph.D. GENENTECH, INC. 1 DNA Way South San Francisco, CA 94080		
6. Total number of applications and patents involved: <u>1</u>		
7. Total fee (37 CFR 3.41): \$ <u>40.00</u> ☐ Enclosed ☒ Authorized to be charged to deposit account		
8. Deposit account number: <u>07-0630</u>		
DO NOT USE THIS SPACE		
9. Statement and signature. To the best of my knowledge and belief, the foregoing information is true and correct and any attached copy is a true copy of the original document. <u>Mark T. Kresnak, Ph.D.</u> Name of Person Signing Registration No. 42,767  Signature <u>8-8-05</u> Date Total number of pages comprising cover sheet, attachments and document: <u>4</u>		

700201259

PATENT
REEL: 016618 FRAME: 0262

345

Aug-08-2005 11:06am From-GENENTECH LEGAL

T-682 P.003/004 F-864

Docket No. P5001R1P2

ASSIGNMENT

WHEREAS,

Patrick Dowd, a citizen of US, residing at 18 Ord Street, San Francisco, CA 94114, and Gretchen Frantz, a citizen of US, residing at 2819 Buchanan Street, San Francisco, CA 94123, and Paul Polakis, a citizen of US, residing at 1449 Cortez Ave, Burlingame, CA 94010, and Victoria Smith, a citizen of AU, residing at 19 Dwight Road, Burlingame, CA 94010, and Susan D. Spencer, a citizen of US, residing at 2395 Paradise Drive, Tiburon, CA 94920, and Thomas D. Wu, a citizen of US, residing at 41 Nevada Street, San Francisco, CA 94110, and Zemin Zhang, a citizen of US, residing at 876 Taurus Drive, Foster City, CA 94404, (hereinafter "ASSIGNORS"), have invented a new and useful invention in

Compositions and Methods for the Diagnosis and Treatment of Tumor

for which an application Serial No. 11/102,501 (Docket No. P5001R1P2) for Letters Patent has been filed by them on April 8, 2005; and

WHEREAS, GENENTECH, INC., a corporation organized and existing under and by virtue of the laws of the State of Delaware, having a place of business at 1 DNA Way, South San Francisco, California 94080-4990, is desirous of acquiring an interest in and to said invention, and in and to the Letters Patents to be obtained therefor;

NOW, THEREFORE, be it known by all whom it may concern;

That for good and valuable consideration the receipt of which is hereby acknowledged, the said ASSIGNORS have and do hereby sell, assign, transfer and set over unto the said GENENTECH, INC., its successors and assigns, the full and exclusive right, title and interest including all rights under the Paris Convention for the Protection of Industrial Property, in and to said invention, and in and to any and all Letters Patents to be granted and issued therefor or any continuation, division, renewal, or substitute thereof, and as to Letters Patents any reissue or re-examination thereof, not only for, to, and in the United States of America, its territories and possessions, but for, to and in all other countries; and it has been and is hereby authorized and requested that the appropriate government agencies issue said Letters Patents to said GENENTECH, INC., in accordance with this Assignment.

Said ASSIGNORS covenant and agree to cooperate with GENENTECH, INC., to enable said GENENTECH, INC. to enjoy to the fullest extent the right, title and interest herein conveyed in the United States and foreign countries. Such cooperation by said ASSIGNORS includes prompt production of pertinent facts and documents, giving of testimony, execution of petitions, oaths, specifications, declarations or other papers, and other assistance all to the extent deemed necessary or desirable by said GENENTECH, INC., (a) for perfecting the right, title and interest herein conveyed; (b) for prosecuting any of said applications; (c) for filing and prosecuting applications for reissuance of any of said patents; (d) for interference or other priority proceedings involving said invention; and (e) for legal proceedings involving said invention and any applications therefor and any patents granted thereon, including without limitation opposition proceedings, cancellation proceedings, priority contests, public use proceedings, infringement actions and court actions; provided, however, that the expense incurred by said ASSIGNORS in providing such cooperation shall be paid for by said GENENTECH, INC.

346

Aug-08-2005 11:06am From:GENENTECH LEGAL

T-682 P.004/004 F-864

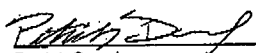
The terms and covenants of this assignment shall inure to the benefit of said GENENTECH, INC., its successors, assigns and other legal representatives, and shall be binding upon said ASSIGNORS, their respective heirs, legal representatives and assigns.

Said ASSIGNORS hereby warrant and represent that they have not entered and will not enter into any assignment, contract, or understanding in conflict herewith.

IN WITNESS WHEREOF we undersign as follows:

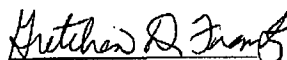
South San Francisco

Dated: 7/14/2005


Patrick Dowd

South San Francisco

Dated: 7/18/2005


Gretchen Frantz

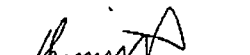
South San Francisco

Dated: 7/15/2005


Paul Polakis

South San Francisco

Dated: 07/15/05


Victoria Smith

South San Francisco

Dated: 7/15/05


Susan D. Spencer

South San Francisco

Dated: 7/15/05


Thomas D. Wu

South San Francisco

Dated: 7/15/05


Zemin Zhang

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Folio 347

2020
Bogotá D.C.,

Doctor(a)
LLOREDA RICAURTE ALICIA
Apoderado(a) y/o Representante de
GENENTECH, INC.

REFERENCIA	Radicación No.	07 104186
	Solicitud internacional	PCT/US2006/013501
	Fecha inicio fase nacional	08/11/2007
	Trámite	11
	Evento	378
	Actuación	416
	Oficio No.	5571
	Folios	1

Teniendo en cuenta que la solicitud de privilegio de patente que se tramita bajo el expediente indicado en la referencia, reúne los requisitos de forma establecidos en la Decisión 486 de la Comisión de la Comunidad Andina, el Tratado de Cooperación en Materia de Patentes (PCT), el Reglamento y la Circular Única de la Superintendencia de Industria y Comercio, se envía el extracto de la solicitud a la Oficina de Comunicaciones para efecto de su publicación en la Gaceta de Propiedad Industrial.

Cúmplase
Dado en Bogotá DC,

29 FEB. 2008

ALIX CARMENZA CESPEDES DE VERGEL
Jefe de la división de Nuevas Creaciones

adpa10