



Geoprocessamento com GRASS-GIS

Carlos Henrique Grohmann

guano@usp.br

Instituto de Energia e Ambiente (IEE) - USP

DOI: 10.6084/m9.figshare.3502184

São Paulo, 26 de julho de 2016

Este documento não é parte da documentação oficial do software descrito. Mais informações podem ser encontradas nos web sites do GRASS-GIS (<http://grass.osgeo.org>) e do QGIS (<http://qgis.org>).

Os detalhes, dados, resultados, etc. descritos neste documento foram escritos e verificados pelo autor; porém, erros quanto ao conteúdo são possíveis. O autor não assume qualquer responsabilidade por falhas e suas consequências. Indicações de possíveis erros são bem-vindas.

Este documento foi escrito em L^AT_EX, e está disponível em formato PDF em <http://carlosgrohmann.com/> ou como fonte L^AT_EX por solicitação ao autor.

Copyright © 2016 Carlos Henrique Grohmann.

É garantida a permissão para copiar, distribuir e/ou modificar este documento sob os termos da Licença de Documentação Livre GNU (GNU Free Documentation License), Versão 1.2 ou qualquer versão posterior publicada pela Free Software Foundation; sem Seções Invariantes, Textos de Capa Frontal, e sem Textos de Quarta Capa. Uma cópia da licença é incluída na seção intitulada “GNU Free Documentation License”.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	vi
1 Introdução	1
1.1 Fontes de informação	1
1.2 Design e estrutura de SIGs	2
1.3 Dimensão de dados em SIG	3
1.4 Estrutura dos comandos no GRASS	5
1.5 Obtendo ajuda com relação aos módulos	5
1.6 Organização dos projetos no GRASS-GIS	6
1.7 O mapset PERMANENT	7
1.7.1 Arquivos do mapset PERMANENT	7
1.8 Organização dos projetos no QGIS	8
2 Mapas Raster - visão geral	9
3 Mapas Vetoriais - visão geral	10
3.1 Manipulação de atributos	12
4 Elementos de cartografia	14
4.1 Escala	14
4.2 Geóide, elipsóide e datum	15
4.3 Sistemas de coordenadas	18
4.4 Coordenadas planas	18
4.5 Coordenadas geográficas (latitude-longitude)	18
4.6 Projeções cartográficas	22
4.7 Projeções cilíndricas	24
4.7.1 Cilíndrica conformal (Mercator)	24
4.7.2 Cilíndrica equidistante	24
4.8 Projeções cônicas	25
4.8.1 Cônica conformal	25
4.8.2 Cônica igual-área	26
4.8.3 Cônica equidistante	26
4.8.4 Policônica	27
4.9 Projeções azimutais	27
4.9.1 Azimutal igual-área	27
4.9.2 Azimutal equidistante	28
4.9.3 Azimutal ortográfica	28
4.9.4 Azimutal gnômica	29
4.10 Projeções compromisso	29

4.10.1	Projeção de Robinson	29
4.11	Coordenadas UTM	30
4.12	Carta Internacional do Mundo ao Milionésimo	31
4.13	Sistema cartográfico nacional	32
4.14	Padrão de exatidão cartográfica	32
4.15	Mapeamento sistemático	33
5	Bem-vindo(a) ao GRASS-GIS!	36
5.1	Criando uma <code>Location</code> e um <code>mapset</code>	36
5.2	Apresentando o <i>Layer Manager</i> e o <i>Map Display</i>	41
6	Dados Raster	43
6.1	Modelos de Elevação SRTM	43
6.1.1	Versões dos dados SRTM	44
6.1.2	Resolução espacial e reamostragem de dados SRTM 03"	47
6.2	Download e importação dos dados	48
6.3	Raster patching	51
6.4	Trabalhando com <code>regions</code>	54
7	Dados vetoriais	61
7.1	Arquivos ESRI shape	61
7.2	Dados vetoriais no QGIS	61
7.3	Arquivos ASCII	66
7.3.1	Geração de pontos aleatórios	67
7.3.2	Amostragem da altitude	67
7.3.3	Exportação como ASCII	70
7.3.4	Importação do arquivo ASCII	70
7.4	Importação de dados vetoriais no GRASS	71
7.5	Buffers	72
8	Trabalhando com máscaras	74
9	Análise Digital de Terreno	76
9.1	Hipsometria	76
9.2	Relevo sombreado	78
9.3	Índices de forma	81
9.4	Análise de vizinhança	84
9.5	Parâmetros do relevo	87
10	Interpolação de superfícies	92
10.1	Reamostragem	92
10.2	Inverso da potência da distância - IDW	93
10.3	Splines regularizadas com tensão - RST	94
10.4	Interpolação bilinear e bicúbica	94
10.5	Outros módulos de interpolação	94

10.6 Reamostragem dos dados SRTM	95
10.7 Extração de curvas de nível	95
10.8 Interpolação de MDE	96
10.8.1 Recorte de dados vetoriais	96
11 Processamento Digital de Imagens	103
11.1 Aquisição das imagens	103
11.2 Importação das imagens no GRASS	106
11.3 Pré-processamento	108
11.4 Índices de vegetação	111
11.5 Classificação	112
12 Visualização n-dimensional	113
13 Produzindo mapas para impressão	115
14 Automatizando tarefas com Python	117
15 Referências Bibliográficas	118

Lista de Figuras

1.1 Geometria e atributos de dados em SIG	3
1.2 Representação do relevo por estrutura discreta e contínua	4
1.3 Dimensão de dados em SIG	4
1.4 Exemplo da estrutura projetos no GRASS-GIS	6
2.1 Resolução espacial de um raster	9
4.1 Exemplos de escalas gráficas.	14
4.2 Representação das ondulações do geóide	15
4.3 Relações entre a superfície da Terra, o geóide e o elipsóide.	15
4.4 Elementos do elipsóide de referência.	16
4.5 Datum topocêntrico (NAD27) e geocêntrico (WGS84).	17
4.6 Sistema de coordenadas planas.	18
4.7 Sistema de coordenadas geográficas (latitude-longitude).	19
4.8 Longitude, latitude geodética e latitude geocêntrica.	19
4.9 Variação do comprimento do grau de longitude com a latitude.	20
4.10 Classificação de projeções cartográficas	23
4.11 Projeções Secantes	23
4.12 Projeção cilíndrica conformal	24
4.13 Projeção cilíndrica equidistante	25
4.14 Projeção cônica conforme de Lambert	25
4.15 Projeção cônica igual-área de Albers	26
4.16 Projeção cônica equidistante	26
4.17 Projeção policônica	27
4.18 Projeção azimutal igual-área de Lambert	27

4.19	Projeção polar azimutal equidistante	28
4.20	Projeção azimutal ortográfica	28
4.21	Projeção azimutal gnomônica	29
4.22	Projeção de Robinson	29
4.23	Divisão das zonas UTM.	30
4.24	Origem das coordenadas nas zonas UTM.	31
4.25	Articulação da Carta Internacional ao Milionésimo em território brasileiro.	32
4.26	Articulação do sistema cartográfico nacional (visão condensada).	34
4.27	Articulação do sistema cartográfico nacional (visão expandida).	35
5.1	Tela de terminal com a mensagem de boas-vindas ao GRASS-GIS.	36
5.2	Janela inicial do GRASS.	36
5.3	Definição do diretório GISBASE e do nome da <code>Location</code> .	37
5.4	Defina como será o método de criação da <code>Location</code> .	37
5.5	Defina a projeção e o sistema de coordenadas.	38
5.6	Escolha do <code>datum</code> WGS-84.	38
5.7	Confira se está tudo certo.	39
5.8	Quase lá.	39
5.9	Defina os limites e a resolução da <code>region</code> padrão.	39
5.10	Defina o nome do seu <code>mapset</code> .	40
5.11	Interface gráfica do GRASS-GIS 7	40
5.12	Janela do <i>Layer Manager</i> .	41
5.13	Barra de ferramentas do <i>Layer Manager</i> .	42
5.14	Barra de ferramentas do <i>Map Display</i> .	42
6.1	Mapa da cobertura dos dados SRTM.	43
6.2	Comparação entre a representação topográfica de MDTs e MDEs.	44
6.3	Mapa da cobertura dos dados SRTM X-SAR para a América do Sul.	46
6.4	Métodos de reamostragem dos dados SRTM utilizados pela NASA e pelo USGS.	47
6.5	Características dos conjuntos de dados SRTM.	48
6.6	Opções do comando <code>r.import</code> .	49
6.7	Insira um novo <i>layer</i> na pilha.	49
6.8	As imagens SRTM, exibidas no <i>Map Display</i> .	50
6.9	Abas Required e Optional do comando <code>r.patch</code> .	51
6.10	Abas Command output e Manual do comando <code>r.patch</code> .	51
6.11	Mapa raster resultante do comando <code>r.patch</code> .	52
6.12	Opções do <i>Map Display</i> .	55
6.13	<i>Map Display</i> com exibição dos limites da <code>region</code> .	56
6.14	Definindo os limites da <code>region</code> a partir da área exibida no <i>Map Display</i> .	56
6.15	Novo mapa raster resultante do comando <code>r.patch</code> .	57
6.16	<code>region</code> ajustada ao limite do <i>raster</i> criado com <code>r.patch</code> .	58
6.17	<code>region</code> ajustada ao limite do dados não-nulos do <i>raster</i> criado com <code>r.patch</code> .	59
6.18	Display com resolução de 30m (esquerda) e 300m (direita).	60
6.19	Display com resolução de 1500m (esquerda) e 3000m (direita).	60
7.1	Interface gráfica do QGIS.	62

7.2	QGIS com layers LM_UF e VETOR_250000 carregados.	63
7.3	Informações sobre um dos polígonos do layer VETOR_250000 (parcial).	63
7.4	Janela de diálogo das propriedades do layer.	64
7.5	Layer VETOR_250000 com transparência no preenchimento.	65
7.6	Janela de informações com dois resultados.	65
7.7	Dados de hidrografia do projeto SIG-Ribeira.	65
7.8	Sistema de coordenadas dos dados do SIG-Ribeira.	66
7.9	Pontos aleatórios gerados com <code>v.random</code>	67
7.10	Mensagem de erro do gerenciador de tabelas de atributos do GRASS.	68
7.11	Opções do comando <code>v.db.addtable</code>	68
7.12	Coluna <code>elev</code> atualizada com os valores do SRTM.	69
7.13	Atributos do ponto selecionado.	69
7.14	Pontos importados com <code>v.in.ascii</code> sobreposto aos com <code>v.random</code>	71
7.15	Dados vetoriais importados somente para a <code>region</code> ativa.	71
7.16	<i>Buffer</i> com distâncias fixas e diferentes com relação a um eixo arbitrário	72
7.17	<i>Buffer</i> com fusão das geometrias (esquerda) e sem fusão (direita).	73
8.1	Salve apenas os limites do PETAR.	74
8.2	Efeito da máscara sobre dados <i>raster</i>	75
8.3	Máscara com dois intervalos de valores (100 thru 300 800 thru 1200).	75
9.1	Reclassificação dos dados SRTM em classes de 100 metros.	77
9.2	Escalas de cores do comando <code>r.colors</code>	77
9.3	Mapa hipsométrico com classes de 100 metros e tabela de cores <code>elevation</code>	78
9.4	Resultado do comando <code>r.relief</code> , com resolução original de 30m.	79
9.5	Resultado do comando <code>r.relief</code> , com resolução de 300m.	80
9.6	Mapa hipsométrico sobreposto a relevo sombreado.	80
9.7	Mapas de declividade e aspecto (orientação de vertentes)	81
9.8	Janela do <code>r.mapcalculator</code>	83
9.9	Mapas aspecto com escala de cores <code>aspect</code> e definida manualmente	84
9.10	Análise por janelas móveis e por janela de busca	85
9.11	Comparação entre análise por janelas móveis e por janela de busca	85
9.12	Comparação de operações de janelas de busca.	86
9.13	Feições morfométricas do relevo (Wood, 1996b)	87
9.14	Resultados do comando <code>r.param.scale</code>	88
10.1	Reamostragem (mudança na resolução espacial) por vizinhos mais próximos.	92
10.2	Reamostragem (transformação geométrica) por vizinhos mais próximos.	93
10.3	Esquema da interpolação por IDW.	93
10.4	Histogramas dos mapas de aspecto	95
10.5	Detalhe das curvas de nível extraídas do SRTM e do SIG-Ribeira	96
10.6	Operações de <i>overlay</i>	97
10.7	Resultado da interpolação por RST (esquerda) e modelo SRTM (direita).	99
10.8	Resultado da subtração entre o modelo SRTM e a interpolação por RST.	100
10.9	Diferenças entre os mapas, exibindo valores maiores que 50m.	101
10.10	Histograma do mapa de diferenças	102

11.1	Determine a cena WRS-2 que recobre a área de estudo.	104
11.2	Ordenação da planilha com as cenas disponíveis.	104
11.3	Cenas disponíveis, ordenadas de acordo com a cobertura de nuvens.	105
11.4	Arquivos da cena LC82200772013315LGN00 (esquerda) e <i>preview</i> (direita).	106
11.5	Bandas dos sensores ETM+ (Landsat 7) e OLI (Landsat 8).	106
11.6	Bandas e comprimentos de onda dos sensores ETM+ (Landsat 7) e OLI (Landsat 8). .	107
11.7	Composição colorida $RBG=4/3/2$	108
11.8	Banda 5 do OLI, antes (esquerda) e depois (direita) do <i>i.landsat.toar</i>	109
11.9	NDVI (esquerda) e EVI (direita).	111
11.10	Classificação não-supervisionada, com 10 classes.	112
12.1	Janelas do <i>Layer Manager</i> com as opções de visualização 3D.	113
12.2	Sobreposição do mapa de diferenças à superfície topográfica.	114
13.1	Mapa gerado pelo <i>ps.map</i>	116

Lista de Tabelas

1	Estrutura dos comandos do GRASS	5
2	Comandos do GRASS para importação de mapas raster	10
3	Comandos do GRASS para importação de mapas vetoriais	12
4	Dimensões de alguns elipsóides comumente utilizados	16
5	Alguns datums comumente utilizados	17
6	Variação dos comprimentos de graus de latitude e longitude	20
7	Nomenclaturas de mapas no Sistema Cartográfico Nacional	34

1. Introdução

Hoje em dia, programas Livres e de código aberto se tornaram sinônimos de inovação e progresso¹. A liberdade de uso, distribuição e modificação dos programas e de seus códigos-fonte asseguram o intercâmbio de idéias entre usuários e desenvolvedores, o que reflete em um sistema estável, com atualizações constantes e orientadas à demanda dos usuários.

O **GRASS-GIS** (*Geographic Resources Analysis Support System*), é um Sistema de Informações Geográficas (SIG), com módulos para processamento e georreferenciamento de imagens multiespectrais, manipulação de dados em formato raster e vetorial, com interfaces para bancos de dados PostgreSQL, MySQL, SQLite, DBF, e ODBC, além de poder ser conectado a UMN/Mapserver, R-stats, gstat, Matlab, Octave, Povray, Paraview e outros programas.

O **QGIS** (*Quantum GIS*) é um SIG com interface amigável e multiplataforma, com suporte a diversos formatos de arquivos raster e vetorial (através das bibliotecas GDAL/OGR) e de bancos de dados. Suporta leitura de arquivos diretamente do GRASS e possui uma *toolbox* que permite rodar diversos comandos do GRASS, funcionando também como uma interface gráfica alternativa à interface nativa do GRASS.

Este tutorial tem como objetivo introduzir o usuário ao uso do GRASS-GIS e do QGIS, com foco principal no GRASS-GIS. Neste tutorial foi utilizado o GRASS-GIS versão 7.0.2 e QGIS versão 2.14 em ambiente Mac OSX; diferenças na interface gráfica podem ocorrer de acordo com a versão do programa e o sistema operacional utilizados. Os exemplos foram produzidos com dados da região do Vale do Ribeira (SP) e podem ser obtidos livremente na Internet.

1.1. Fontes de informação

Algumas fontes de informação, manuais e tutoriais que podem ajudar a aprender e usar o GRASS-GIS e o QGIS.

- GRASS-GIS web site oficial: <http://grass.osgeo.org>
- GRASS Wiki <http://grass.osgeo.org/wiki/>
- Lista de discussão eletrônica: <http://grass.osgeo.org/community/support.php>
- QGIS web site oficial: <http://qgis.org>
- QGIS Wiki <http://www.qgis.org/wiki/>
- Lista de discussão eletrônica: <http://qgis.org/en/community/mailing-lists.html>

¹Leia mais sobre Software Livre em <http://www.gnu.org>

- Neteler & Mitasova (2008). *Open Source GIS: A GRASS GIS Approach*. Boston, Kluwer Academic Publishers.

1.2. Design e estrutura de Sistemas de Informação Geográfica

De maneira geral, um Sistema de Informações Geográficas pode ser caracterizado por quatro componentes principais:

- Entrada de dados
- Administração
- Análise
- Apresentação

Os tipos de dados nesse sistema de quatro componentes podem ser classificados em **geométricos, atributos e gráficos**:

Dados geométricos descrevem a situação espacial dos objetos, sua forma e sua posição no espaço. As relações entre pontos individuais, linhas e áreas são dadas por um sistema de coordenadas de referência, e sua relações com o mundo real. Dados geométricos podem ser do tipo raster (matricial, *pixel*) ou vetorial (pontos, linhas, polígonos, áreas, sólidos - ver figura 1.1):

Dados raster são dados regularmente espaçados no espaço, em uma estrutura de matriz com células quadradas (normalmente) e de mesmo tamanho. Este tipo de dado é utilizado em imagens de satélite, modelos de relevo e mapas escaneados, por exemplo.

Dados vetoriais são usados para informações de objetos lineares (linhas e polilinhas) ou áreas definidas por linhas fechadas (polígonos).

Dados pontuais podem ser considerados uma forma especial de dados vetoriais (pontos vetoriais), e são usados para armazenar informações pontuais espalhadas em uma área, como pontos de coletas de amostras, localização de sítios arqueológicos, etc.

Atributos (*categorias*) são, como o nome sugere, atributos relacionados aos tipos de dados mencionados anteriormente, e normalmente são armazenados em um banco de dados interno ao sistema SIG ou em um banco de dados externo através de uma interface tipo DBMI (*Database Management Interface*).

Gráficos descrevem os métodos usados para desenhar os objetos espaciais nos vários dispositivos de saída existentes (tela do monitor, impressora, etc).

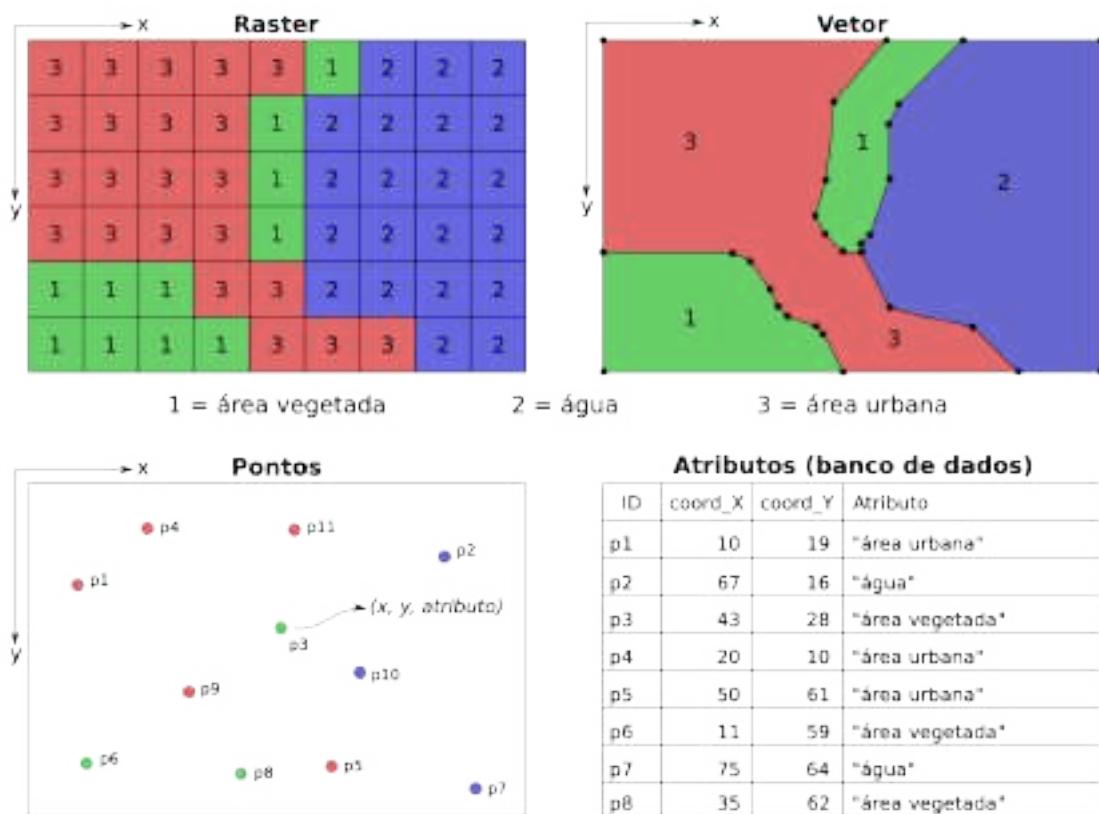


Figura 1.1. Geometria e atributos de dados em SIG

Em um SIG, as relações entre esses tipos de dados (geométricos, atributos e gráficos) definem duas **estruturas** básicas de dados:

- Estruturas **contínuas** → objetos laminares e ilimitados no espaço (superfícies)
- Estruturas **discretas** → objetos definidos como linhas e áreas

Em um SIG, módulos específicos permitem a conversão entre estruturas diferentes. Curvas de nível, por exemplo, podem ser armazenadas como vetores (estruturas discretas) e podem ser convertidos para uma representação contínua do terreno (raster) por interpolação. Nesse caso a qualidade do modelo resultante irá depender da qualidade e resolução dos dados originais (Fig. 1.2).

1.3. Dimensão de dados em SIG

Dados espaciais podem ser representados por *pontos*, *linhas*, *áreas/superfícies* e *volumes* (Fig. 1.3).

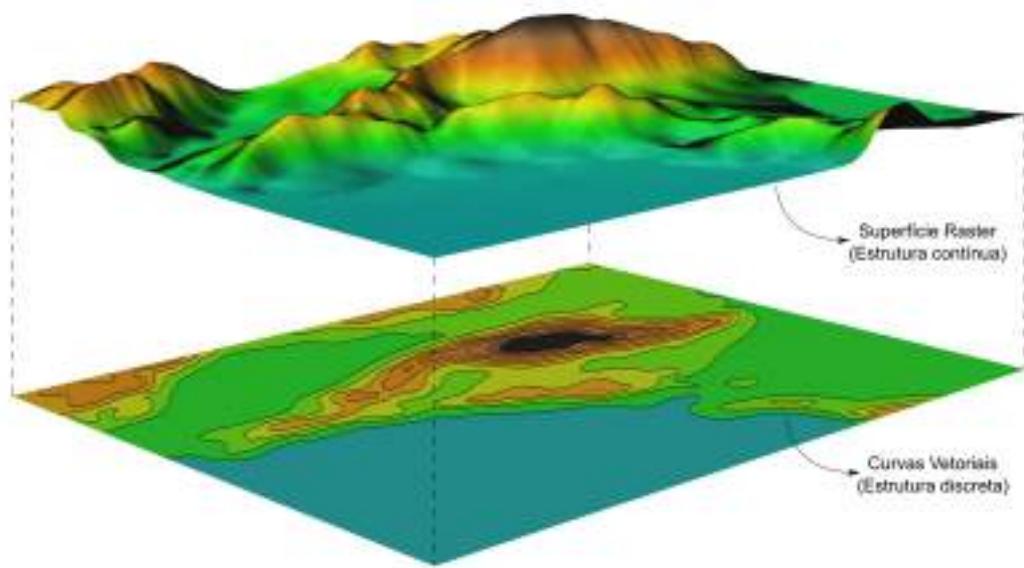


Figura 1.2. Representação do relevo de uma área por estrutura discreta (curvas de nível vetoriais) e contínua (superfície raster gerada por interpolação).

Pontos podem representar a localização de objetos reais (como uma casa) ou virtuais (como o epicentro de um terremoto), e não têm dimensão física. Linhas têm apenas uma dimensão física, o comprimento, enquanto que as áreas têm duas dimensões (2D: comprimento e largura). No caso de superfícies contínuas, como em um modelo da superfície topográfica, onde há um terceiro parâmetro além das coordenadas X e Y (Z=elevação, por exemplo), dizemos que os dados têm 2,5D. Isto pode parecer um pouco estranho, já que os programas de SIG nos permitem visualizar as superfícies em perspectivas tri-dimensionais, porém apesar do parâmetro Z ser utilizado para gerar um imagem 3D, a superfície não possui um valor de espessura (a terceira dimensão física). Os únicos dados realmente tri-dimensionais são aqueles que possuem quatro parâmetros (XYZ + atributo) como no caso de sólidos representando edifícios ou em modelos que utilizam *voxels*, que são o equivalente dos pixels em três dimensões, representados por cubos no espaço contendo um valor de atributo .

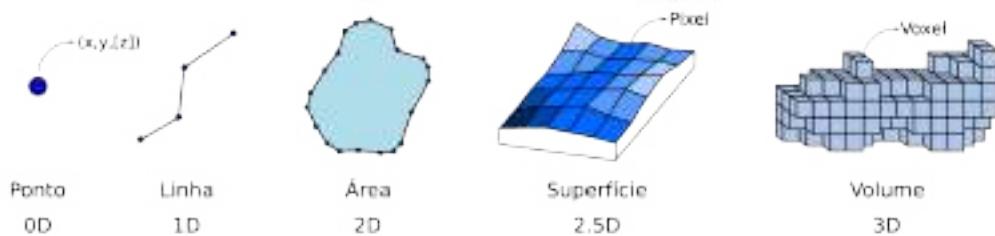


Figura 1.3. Dimensão de dados em SIG

1.4. Estrutura dos comandos no GRASS

O GRASS-GIS é um Sistema de Informações Geográficas com funções orientadas a dados do tipo raster ou vetoriais. Cada função é conduzida por um módulo próprio, o que faz do GRASS um SIG bem estruturado e transparente. Além disso, como apenas os módulos necessários estão em execução, os recursos do computador são preservados, e garantem um melhor desempenho do sistema.

A estrutura dos comandos do GRASS é bastante simples e clara. Os comandos (chamados de *módulos* no GRASS) tem nomes auto-explicativos e seu tipo pode ser reconhecido pelo prefixo. Por exemplo, o módulo para obter informações sobre mapas *raster* é **r.info** e o módulo para converter de formato vetorial para raster é o **v.to.rast**. Na tabela 1 vemos a estrutura geral dos nomes dos módulos em mais detalhe.

Além dos módulos existentes no GRASS, todos os programas do ambiente Unix/Linux estão disponíveis com uso de linhas de comando na janela de terminal (ou de console). Apesar de muitos usuários iniciantes considerarem difícil usar estas funcionalidades, a programação de *shell scripts* é simples de se aprender, e pode ser usada para automatizar tarefas ou extender as funcionalidades do ambiente SIG de acordo com as necessidades de cada um.

Prefixo	Função	Significado dos comandos
d.*	display	Exibição e consulta gráfica no Map Display
r.*	raster	Processamento de dados raster
i.*	imagem	Processamento de imagens
v.*	vetor	Processamento de dados vetoriais
g.*	geral	Comandos gerais para manutenção de arquivos
ps.*	postscript	Criação de mapas em formato postscript
db.*	database	Gerenciamento de banco de dados
r3.*	voxel	Processamento de dados raster 3D

Tabela 1. Estrutura dos comandos do GRASS

1.5. Obtendo ajuda com relação aos módulos

Arquivos de ajuda estão disponíveis para quase todos os 400 módulos do GRASS, onde são descritas a funcionalidade do módulo e a sintaxe de uso. Uma ajuda breve pode ser obtido com o uso do parâmetro **-help**, na linha de comando, como em **d.rast -help**. Descrições mais detalhadas encontram-se nas páginas de manual para cada módulo. Essas páginas podem ser acessadas on-line no web site do GRASS ou com o comando **g.manual**, como em **g.manual d.rast**, ou clicando no botão **Help** na janela de cada comando. A página de manual será aberta em um navegador de internet.

1.6. Organização dos projetos no GRASS-GIS

A organização dos projetos no GRASS-GIS segue uma hierarquia baseada em **Locations** e **mapsets**. A **Location** comprehende toda a área de trabalho do projeto, enquanto que o **mapset** é a porção ativa e utilizada para análise, que pode ser do mesmo tamanho ou menor que a **location**. Ainda, vários **mapsets** podem ser definidos para a mesma **location**.

Em ambientes multiusuário, várias pessoas podem trabalhar na mesma **Location** ao mesmo tempo, mas não no mesmo **mapset**. Para contornar essa limitação, dados de interesse comum (tais como modelos de relevo, imagens de satélite etc) podem ser armazenados em um **mapset** especial ao qual todos os usuários têm acesso, chamado de **PERMANENT**, que é criado automaticamente ao se criar uma nova **Location**.

Um conceito importante dentro do GRASS é o de **region**, que define, dentro do **mapset**, a área de interesse e a resolução espacial dos mapas *raster*. Tanto a resolução espacial quanto as coordenadas do retângulo envolvente da **region** podem ser facilmente alteradas sem a necessidade de reinicialização do sistema ou a criação de novos projetos; é possível salvar as configurações da **region** para acessá-la facilmente quando necessário.

É preciso frisar que todas as análises envolvendo mapas *raster* (análise de terreno, álgebra de mapas, interpolação de superfícies etc) são efetuadas de acordo com as configurações da **region** ativa, e que esta não necessariamente corresponde com as configurações do *Display*.

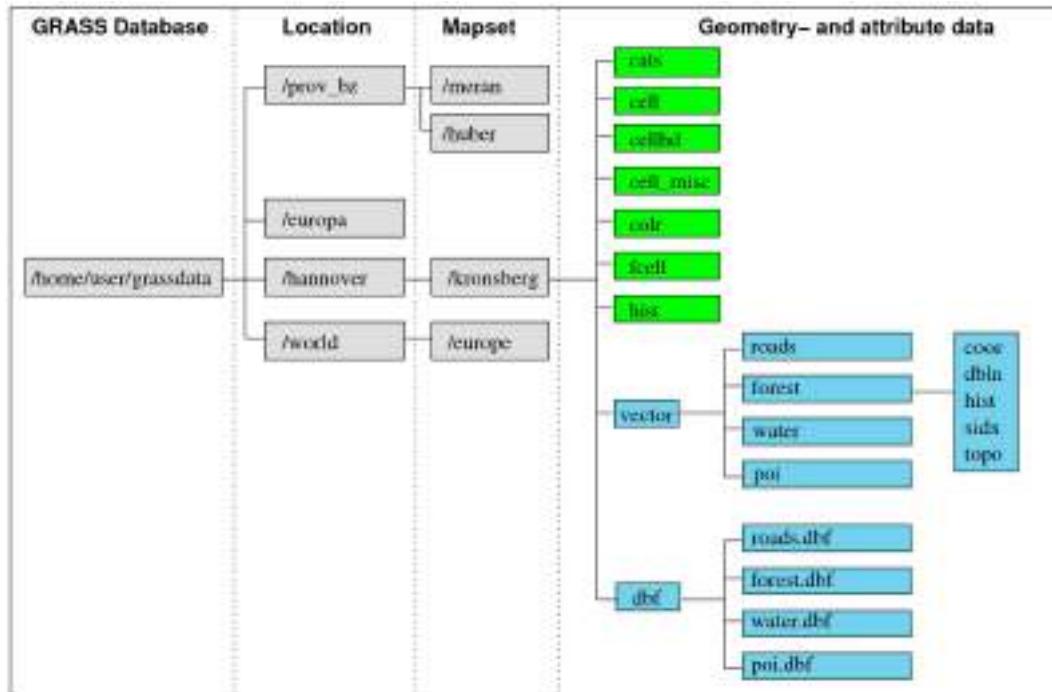


Figura 1.4. Exemplo da estrutura de projetos no GRASS-GIS, extraído de Dassau *et al.* (2005).

Os dados referentes aos projetos são armazenados em um diretório (pasta) chamado pelo GRASS de **GISBASE**. As **Locations** serão subdiretórios de **GISBASE**, os **mapsets** serão subdiretórios dentro de cada **Location** e assim por diante.

Normalmente o diretório **GISBASE** (p.ex., **grassdata**) é criado antes de se rodar o GRASS pela primeira vez:

```
> cd <Enter>
> mkdir grassdata <Enter>
```

Em sistemas unix, o comando **cd** (*change directory*) leva ao diretório inicial do usuário. Com **mkdir** (*make directory*), o diretório **grassdata** é criado.

Uma vez que cada parte dos *layers* (p.ex., geometria, tabela de atributos, objetos gráficos) é armazenada em subdiretórios diferentes, a manutenção dos dados dos projetos (p.ex., copiar, apagar, renomear) deve ser feita apenas com os comandos apropriados (**g.copy**, **g.remove**, **g.rename**).

1.7. O mapset PERMANENT

Todas as informações sobre projeção cartográfica, resolução espacial e extensão da área do projeto são armazenadas no **mapset** **PERMANENT**, que é gerado automaticamente pelo GRASS ao se criar uma **Location**. Se necessário dados de interesse comum podem ser armazenados neste **mapset**. Como apenas o usuário que criou a **Location** tem permissão de escrita ao **mapset** **PERMANENT**, tem-se a garantia de que os dados não serão apagados ou modificados acidentalmente. Outros usuário (e também a pessoa com permissão de escrita ao **mapset** **PERMANENT**) devem criar **mapsets** adicionais para criar, salvar e alterar seus próprios arquivos e análises baseadas nos dados do **mapset** **PERMANENT**.

1.7.1 Arquivos do mapset PERMANENT

Se nenhum mapa for armazenado em **PERMANENT**, o diretório irá conter apenas informações sobre o projeto:

- **DEFAULT_WIND** – Especificações dos limites e resolução espacial da **region** padrão (default);
- **MYNAME** – Nome do projeto. p.ex., doutorado, relevoAmazonia;
- **PROJ_INFO** – Especificações sobre a projeção cartográfica, *datum* e elipsóide;
- **PROJ_UNITS** – Unidade de medidas (metros, graus);
- **WIND** – Especificações dos limites e resolução espacial da **region** atual.

1.8. Organização dos projetos no QGIS

No QGIS não é necessário importar arquivos. Os dados são acessados diretamente, em arquivos locais ou via rede, pela biblioteca GDAL/OGR. A [des]organização dos dados fica a cargo do usuário, tal como no ArcGIS ou gvSIG.

O QGIS também é capaz de realizar projeções entre datums e sistemas de coordenadas em tempo real (projeção *on-the-fly*), não sendo necessário converter todos os dados para um datum e sistema de coordenadas único, como no caso do GRASS. Ao mesmo tempo em que a projeção *on-the-fly* é muito útil, a obrigatoriedade da conversão dos dados para um referencial cartográfico único é vista por muitos como benéfica, pois minimiza a possibilidade de problemas nas análises.

2. Mapas Raster - visão geral

Mapas raster são compostos por células, sendo que cada célula (ou *pixel*: *picture element*) recebe o valor de um atributo, que representa um fenômeno (por exemplo temperatura ou altitude). A *resolução espacial* de um raster equivale à dimensão da célula no mundo real. Quanto maior a resolução espacial, maior é o nível de detalhe que o raster pode exibir, e portanto menor o tamanho do pixel no mundo real (Fig. 2.1).

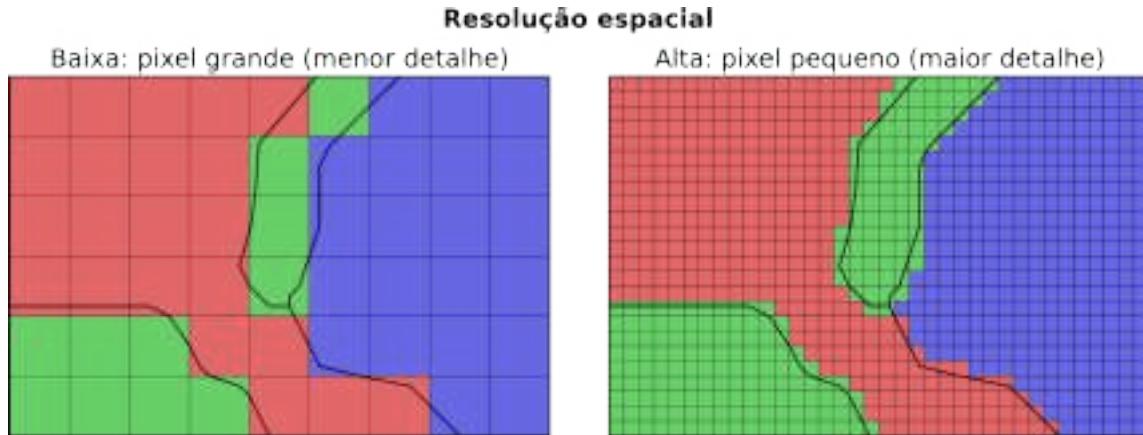


Figura 2.1. Resolução espacial de um raster

As células são organizadas em linhas e colunas, e seu valor pode ser acessado pelas coordenadas absolutas da matriz (linha/coluna) ou pelas coordenadas geográficas. Mapas raster são normalmente utilizados para aplicações de sensoriamento remoto – como análise de imagens orbitais ou fotografias aéreas – e para interpolação de dados irregularmente distribuídos no espaço, tal como a geração de Modelos Digitais de Terreno (MDT) a partir de pontos coletados por GPS ou por levantamentos a laser aeroportados (LiDAR).

Uma das desvantagens de se trabalhar com dados raster é a quantidade de memória necessária e o tempo de processamento das análises, que cresce exponencialmente com o aumento da resolução espacial utilizada (particularmente no caso de modelos de elevação de alta resolução como os levantamentos a laser), porém com o contínuo aumento na capacidade de processamento e de armazenamento dos computadores atuais, essa questão já não é mais considerada como essencial. Outro ponto a ser lembrado é que dados raster não possuem relações de vizinhança (Topologia: o polígono A está à direita do polígono B, e assim por diante), uma vez que cada pixel é definido de maneira independente dos outros.

No GRASS é preciso importar arquivos raster, que serão convertidos para um formato próprio. No QGIS, os arquivos são acessados diretamente, sem necessidade de importação. O GRASS pode importar diversos tipos de mapas raster. De maneira geral, podemos agrupá-los em três categorias:

Imagen: Rasters em formato de imagem sempre possuem valores positivos e inteiros, tais como os formatos PPM, PNG, JPEG, e GIF.

ASCII: O formato ASCII podem conter valores positivos, negativos, inteiros ou decimais. Os arquivos ASCII-GRID do ArcInfo são exemplos desse formato.

Binário: Rasters binários podem ter valores inteiros ou decimais, positivos ou negativos, que podem estar em diferentes canais (bandas), em diferentes resoluções espaciais. Exemplos são os arquivos [Geo]TIFF e ERDAS/IMG.

Ao se importar um mapa raster no GRASS, deve-se sempre lembrar que os mapas são importados com sua extensão e resolução originais, mas exportados de acordo com a extensão e resolução definidas pela `region` ativa.

A tabela 2 lista os comandos usados para importar diferentes formatos raster:

Comando do GRASS	Formato Raster
<code>r.in.ascii</code>	GRASS ASCII
<code>r.in.bin</code>	BIL, arquivos binários do GMT, LANDSAT TM5/ETM+
<code>r.in.gdal</code>	ArcInfo ASCII/Binary GRID, BIL, ERDAS (LAN, IMG), USGS DOQ, JPEG, SAR CEOS, EOSAT, GeoTIFF, PPM/PNM, SDTS DEM, GIF, PNG
<code>r.in.srtm</code>	arquivos SRTM em formato HGT (usa GDAL)
<code>r.in.arc</code>	ESRI ARC/INFO ascii raster
<code>r.in.aster</code>	Imagens Terra-ASTER e ASTER DEMs

Tabela 2. Comandos do GRASS para importação de mapas raster

O módulo mais utilizado para importação (e exportação) de rasters é certamente o `r.in.gdal`, que usa a biblioteca GDAL (*Geospatial Data Abstraction Library*²).

3. Mapas Vetoriais - visão geral

Mapas vetoriais representam linhas, polilinhas (*polylines*) ou polígonos.

Uma linha conecta dois pontos extremos (*end points* – nós ou vértices), cada um com suas respectivas coordenadas XY ou XYZ. Cada objeto vetorial pode possuir vários (ou nenhum) atributos, armazenados em um banco de dados. Para uso interno ao GRASS, os bancos de dados *dBase* e *sqlite* estão disponíveis. Existem interfaces para acesso à DBMS (*Database Management System*) externos, como PostgreSQL, Oracle, mySQL, Access, etc. Para o armazenamento externo ao GRASS de dados geométricos e atributos, existem interfaces para PostGIS ou Oracle Spatial.

²<http://www.gdal.org/>

Em comparação com dados raster, os dados vetoriais utilizam menos memória e têm menor tempo de processamento na maioria das análises. Além disso, no GRASS, os vetores possuem relações de topologia, o que significa que cada objeto vetorial “sabe” quais vértices possui e/ou com quais áreas faz vizinhança.

Mais precisamente, isso significa que elementos de mapas vetoriais compartilham componentes. Por exemplo, em um sistema não topológico, se duas áreas (polígonos) possuem uma borda (aresta) em comum, essa borda teria que ser digitalizada e armazenada duas vezes, o que não ocorrem em um sistema topológico, onde a borda em comum é compartilhada pelas duas áreas. Esse tipo de representação ajuda a manter mapas com uma geometria mais “limpa” e permite análises que não podem ser realizadas com dados não topológicos (também chamados de vetores tipo “espaguete”, uma vez que cada polígono tem que ser digitalizado inteiro). No GRASS-GIS, vetores com topologia são referidos como de **nível 2**, e dados tipo espaguete são de **nível 1**.

Entretanto, a informação de topologia nem sempre é necessária, e os requisitos de memória e espaço em disco podem se tornar impraticáveis com determinados tipos de dados (como os resultantes de levantamentos topográficos por laser aeroportados - LiDAR). Sendo assim, existem dois módulos do GRASS que permitem trabalhar com dados vetoriais de nível 1 (sem topologia). O módulo `v.in.ascii` é usado para importação de dados pontuais e pode criar um layer vetorial sem construir a tabela de topologia. Normalmente é usado em conjunto com `v.surf.rst`, que realiza a interpolação de valores a partir de pontos ou isolinhas.

Os seguintes objetos vetoriais estão definidos no GRASS-GIS:

ponto: um ponto no espaço;

linha: uma seqüência de pontos (vértices) conectados, com dois pontos extremos (*endpoints*) chamados de nós (*nodes*);

borda (boundary): a linha que define uma área;

centróide: um ponto dentro de uma borda fechada;

área: a composição topológica de *borda* + *centróide*;

face: uma área tridimensional;

núcleo (kernel): um centróide tridimensional dentro de um volume (ainda não implementado);

volume: um corpo tridimensional, a composição topológica das *faces* + *núcleo* (ainda não implementado);

O módulo `v.type` pode ser usado para conversão entre tipos vetoriais (se possível). O módulo `v.build` é usado pra gerar topologia, e opcionalmente permite a extração de objetos espúrios em um novo mapa. Erros de topologia podem ser corrigidos manualmente dentro do ambiente do módulo `v.digit` ou, até certo ponto, automaticamente com `v.clean`. Polígonos adjacentes pode ser detectados com `v.to.db` (opção '`sides`').

Ao se importar um mapa vetorial, é preciso considerar que existem diversos formatos e padrões diferentes de arquivos. A tabela 3 lista os comandos usados para importação de dados vetoriais. Provavelmente o formato mais utilizado é o ESRI SHAPE (o famoso *shapefile*). O módulo `v.in.ogr` é o mais utilizado, por oferecer uma interface com a maioria dos formatos vetoriais. Adicionalmente, esse módulo tem opções para criar novas `Locations` ou estender a `region` padrão de acordo com os limites do mapa a ser importado. Outros módulos existem para casos mais específicos, como o `v.in.ascii`, usado para importar dados de arquivos texto com as coordenadas e os atributos de pontos no espaço (como no caso de dados LiDAR), e o `v.in.db`, para importar a partir de um banco de dados contendo as coordenadas e atributos dos dados.

O módulo `v.external` permite que mapas externos sejam ligados virtualmente ao `mapset`, não necessitando de importação (e consequente conversão para o formato nativo do GRASS). Apenas uma pseudo-topologia é gerada e a geometria do vetor não é importada. Os comandos `v.out.*` são usados para exportar os mapas vetoriais em vários formatos.

Comando do GRASS	Formato Vetorial
<code>v.in.ogr</code>	SHAPE file, UK.NTF, SDTS, TIGER, S57, MapInfo-File, DGN, VRT, AVCBin, REC, Memory, GML, ODBC (veja também: http://www.gdal.org/ogr/ogr_formats.html)
<code>v.in.ascii</code>	Arquivos texto em formato <code>x y z</code> e vetores ASCII exportados por versões antigas do GRASS
<code>v.in.dxf</code>	Arquivos DXF (CAD Desktop eXchange Format)
<code>v.in.garmin</code>	Importa <i>Waypoints</i> , <i>Tracks</i> e <i>Routes</i> de um GPS Garmin
<code>v.in.e00</code>	ArcInfo-E00
<code>v.in.db</code>	Cria vetores a partir de bancos de dados com coordenadas <code>x y z</code>

Tabela 3. Comandos do GRASS para importação de mapas vetoriais

3.1. Manipulação de atributos

Após a importação, o mapa estará disponível no formato vetorial nativo do GRASS. A geometria, topologia e os atributos são armazenados em um banco de dados interno do GRASS (em formato SQLITE por padrão a partir da versão 7). O GRASS pode ser conectado a um ou vários bancos de dados. Os comandos `bd.*` fornecem funcionalidades básicas de SQL (*Structured Query Language*) para manutenção de atributos, enquanto que os comandos `v.db.*` operam nos mapas vetoriais,

- Categorias: o número da categoria é o **ID** (IDentificador) do vetor e é usado para conectar (link) atributo(s) à cada objeto vetorial (cada objeto vetorial pode ter zero, uma, duas ou mais categorias). Os números de categoria são armazenados tanto no arquivo de geometria quanto na(s) tabela(s) de atributos para cada objeto vetorial (normalmente é a coluna “`cat`”). A visualização e manutenção dos números de categoria pode ser feita com `v.category`. Para poder ligar um vetor à mais de uma tabela de atributos, mais de um número de categoria é necessário.
- Layers (camadas): é possível ligar (*link*) os objetos geográficos em um mapa vetorial a uma ou mais tabelas. Cada *link* com uma tabela diferente é chamado de **layer**. Um link define qual o *driver* de acesso ao banco de dados (MySQL, PostgreSQL, etc), qual o banco de dados e qual tabela desse banco de dados será utilizada. Cada número de categoria no arquivo de geometria corresponde à um linha na tabela de atributos (a coluna de ligação é normalmente a coluna “`cat`”). A visualização e manutenção dos layers pode ser feita com `v.db.connect`.

Os layers do GRASS não contém nenhum objeto geográfico, mas na verdade consistem de links para tabelas de atributos nas quais os objetos podem ter zero, uma ou mais categorias. Se um objeto vetorial tem zero categorias em um layer, então ele não é exibido nesse layer. Os objetos vetoriais podem, portanto, aparecer em um determinado layer, mas não em outro(s). Esse sistema permite que objetos topologicamente conectados, mas tematicamente distintos (como florestas e lagos, por exemplo) sejam colocados em um mesmo mapa. Outra possibilidade é a ligação de atributos temporais à localidades que não mudam com o passar do tempo.

Por padrão, o primeiro layer é ativo, ou seja, a primeira tabela corresponde ao primeiro layer. Outras tabelas são ligadas aos layers seguintes.

- Suporte à SQL: Comandos SQL podem ser executados diretamente com `db.execute`, `db.select` e outros módulos `db.*`.

4. Elementos de cartografia

Antes de criar um projeto é preciso definir qual projeção cartográfica será utilizada. Além disso, no GRASS não é possível fazer transformações de coordenadas *on-the-fly*, ou seja, utilizar dados em diferentes sistemas de coordenadas ao mesmo tempo. Nas seções seguintes veremos brevemente alguns conceitos de geodésia, cartografia e os parâmetros das projeções mais utilizadas.

4.1. Escala

A escala de um mapa define a relação entre a medida de um objeto ou lugar representado no papel e sua medida real:

$$E = \frac{d}{D}$$

onde d é a distância medida no mapa e D a distância real. A escala pode ser representada em sua forma *numérica*, como em $1/10.000$ e $1:10.000$ ou de forma *gráfica*. Uma escala de $1:10.000$ significa que os objetos e distâncias reais são 10.000 vezes maiores do que suas representações em mapa (ou seja, 1 cm no mapa equivale a 100 m – ou 10.000 cm – no terreno).

A escala gráfica (fig. 4.1) é uma representação de várias distâncias do terreno sobre uma linha reta graduada, de acordo com a unidade de medida utilizada (quilômetros, milhas, etc). O segmento à direita do valor zero é chamado de *escala primária*, e o segmento à esquerda do zero recebe o nome de *talão* ou *escala de fracionamento*, e é divido em sub-múltiplos da unidade de medida utilizada, graduado da direita para a esquerda.

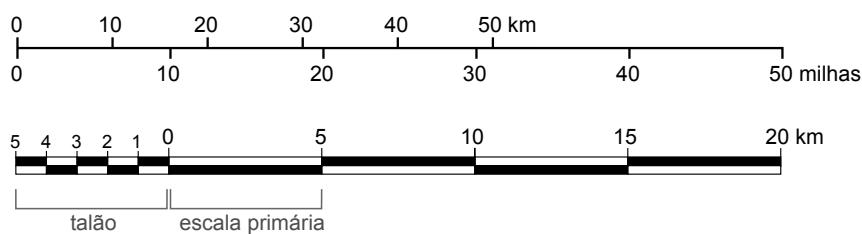


Figura 4.1. Exemplos de escalas gráficas.

A escala é uma divisão matemática entre os valores real e em mapa, portanto uma escala de $1:1.000$ é maior que $1:100.000$:

$$\frac{1}{1000} = 0.001 > \frac{1}{100000} = 0.00001$$

Em outras palavras, quanto maior a escala do mapa, maior o nível de detalhe apresentado.

4.2. Geóide, elipsóide e datum

O geóide é considerado como uma representação matemática bastante precisa da forma da Terra (fig. 4.2). É a superfície equipotencial do campo gravitacional da Terra, que coincide com o nível médio dos oceanos (fig. 4.3). A determinação dos parâmetros do geóide é alvo de estudo da geodésia, a ciência que trata do levantamento e da representação da forma e da superfícies da Terra, incluindo o campo gravitacional. Devido à complexidade matemática do geóide, a forma da Terra é geralmente representada por elipsóides.

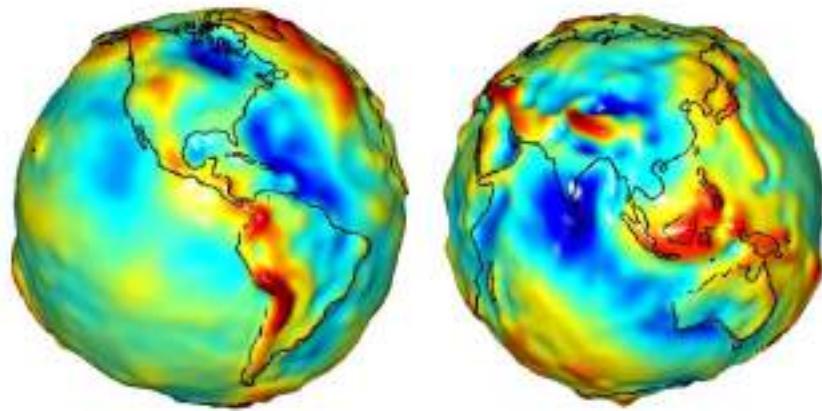


Figura 4.2. Representação das ondulações do geóide, construída a partir de dados de anomalia gravimétrica (NASA/JPL/University of Texas Center for Space Research).

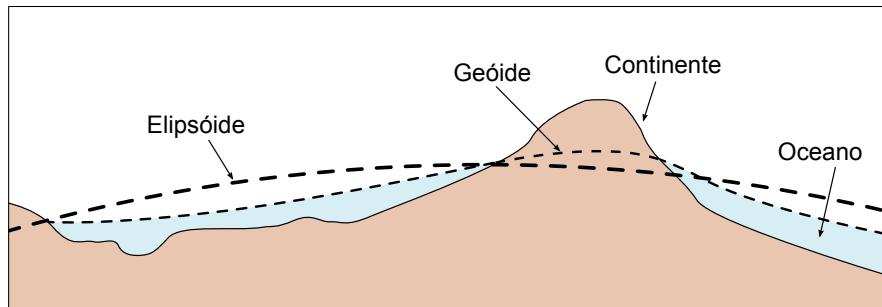


Figura 4.3. Relações entre a superfície da Terra, o geóide e o elipsóide.

Como a simplificação da forma da Terra para uma esfera não é preciso a suficiente para gerar mapas em escala maior (com maior detalhe) do que 1:2.000.000, é necessário utilizar *elipsóides de revolução*, que possuem achatamento nos pólos, para tentar definir uma superfície mais simples de se trabalhar do que o geóide, e ainda assim precisa o suficiente para fins cartográficos.

Matematicamente, o elipsóide de referência é normalmente um esferóide oblato (achatado) com dois eixos diferentes, o raio equatorial ou semi-eixo maior (a) e o raio polar ou semi-eixo menor (b). As relações entre os semi-eixos definem as propriedades de achatamento (*flattening*, f) e excentricidade (e):

$$f = \frac{(a - b)}{a} = 1 - \frac{(b)}{a}$$

$$e = \frac{\sqrt{a^2 - b^2}}{a}$$

Usualmente, os elipsóides são definidos em termos do semi-eixo maior (a) e do inverso do achatamento ($1/f$). A figura 4.4 ilustra os elementos do elipsóide de referência e a tabela 4 lista os parâmetros de alguns elipsóides comumente utilizados.

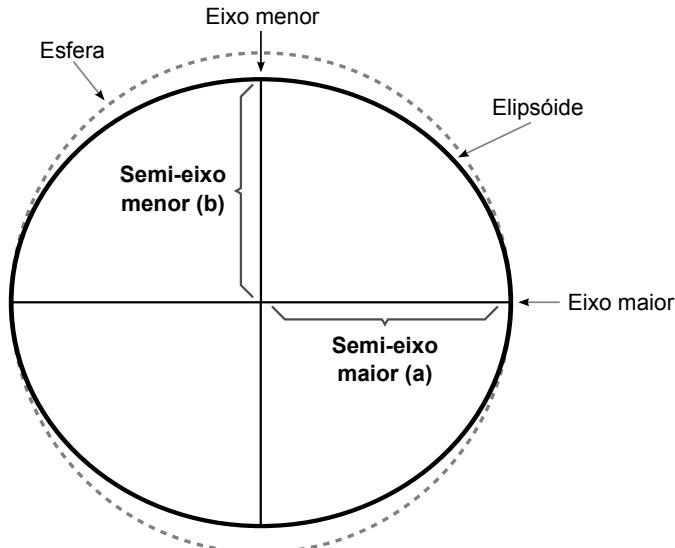


Figura 4.4. Elementos do elipsóide de referência.

Elipsóide	Semi-eixo maior a (m)	Semi-eixo menor b (m)	Achatamento ($1/f$)
Clarke 1866	6 378 206,4	6 356 583,8	1/294,9786982
Internacional 1924	6 378 388,0	6 356 911,9	1/297,0
GRS 1980	6 378 137,0	6 356 752,3141	1/298,257222101
WGS 1984	6 378 137,0	6 356 752,3142	1/298,257223563
SGR 1967	6 378 160,0	6 356 776,0	1/298,25
Esfera (6371 km)	6 371 000,0	6 371 000,0	∞

Tabela 4. Dimensões de alguns elipsóides comumente utilizados

Um *datum* (no plural, *datums* ou *data*), é uma referência a partir da qual são realizadas medidas. Em cartografia datums são utilizados para definir sistemas de coordenadas, sejam eles *globais* ou *locais*. Existem datums *topocêntricos*, cuja origem é um ponto na superfície terrestre, e datums *geocêntricos*, com origem no centro de massa da Terra.

Na figura 4.5 o datum NAD27 (*North American Datum 1927*) é do tipo topocêntrico, e define um sistema de coordenadas *local*. Veja que o elipsóide se ajusta bem à área da América do Norte, mas não se ajusta à região da Austrália. O datum WGS84 (*World Geodetic System 1984*) é do tipo geocêntrico, e busca um ajuste médio ao planeta. Por definir um sistema de coordenadas *global*, é utilizado no mundo todo.

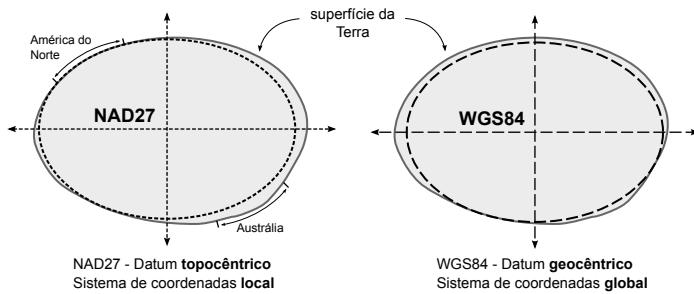


Figura 4.5. Datum topocêntrico (NAD27) e geocêntrico (WGS84).

O datum SAD-69 (*South American Datum 1969*), amplamente utilizado no Brasil, tem sua origem no vértice do Sistema Geodésico Brasileiro localizado no município de Chuá (MG). Recentemente completamos uma fase de transição do datum SAD-69 (topocêntrico) para o novo datum oficial do Brasil, o SIRGAS³ (*Sistema de Referência Geocêntrico para a América do Sul*), que como o nome indica, trata-se de um datum geocêntrico.

O sistema de referência do SIRGAS coincide com o IERS⁴ (*International Earth rotation and Reference systems Service*) e com o ITRF⁵ (*International Terrestrial Reference Frame*), e tem os parâmetros do elipsóide *Geodetic Reference System* (GRS-80), sendo considerado idêntico ao WGS-84 para fins cartográficos. A tabela 4 lista as características de alguns datums comumente utilizados.

Datum	Região de uso	Origem	Elipsóide
WGS 84	Global	Centro de massa da Terra	WGS 84
NAD 83	América do Norte, Caribe	Centro de massa da Terra	GRS 80
ED 50	Europa, África (norte)	Potsdam	Internacional 1924
SAD 69	América do Sul	Chuá	SGR 67
SIRGAS	América da Sul	Centro de massa da Terra	GRS 80

Tabela 5. Alguns datums comumente utilizados

³<http://www.ibge.gov.br/sirgas/>

⁴<http://www.iers.org/>

⁵<http://itrf.ensg.ign.fr/>

4.3. Sistemas de coordenadas

De modo geral, podemos agrupá-los em sistemas globais e locais. Sistemas globais requerem coordenadas esféricas, enquanto que sistemas locais podem trabalhar com coordenadas planas em diferentes unidades de medida (metros, quilômetros, pés, milhas etc).

4.4. Coordenadas planas

O sistema de coordenadas planas é o mais simples de utilizar. Assim como em um gráfico cartesiano tipo XY, a origem dos dados é determinada pelo cruzamento de dois eixos perpendiculares e os valores (coordenadas) aumentam para a direita (em X) e para cima (em Y). O eixo horizontal é chamado de abcissa, e o eixo vertical, de ordenada (fig. 4.6).

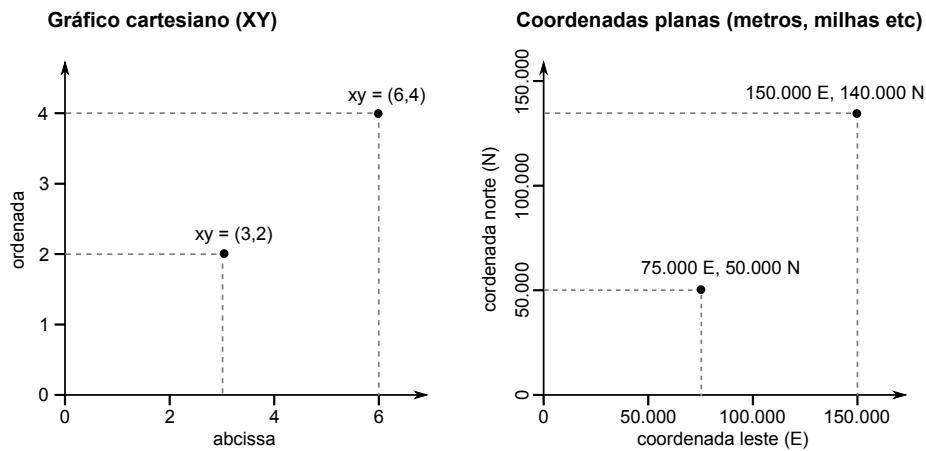


Figura 4.6. Sistema de coordenadas planas.

4.5. Coordenadas geográficas (latitude-longitude)

Neste sistema os planos de referência são o Equador e o Meridiano de Greenwich. A Terra é dividida em 180° de *longitude* a partir de Greenwich para Leste e para Oeste, e em 90° de *latitude* para Sul e Norte do Equador (fig. 4.7). A altitude é medida a partir do centro de massa da Terra. Em geodésia, a latitude é representada pela letra grega *phi* (ϕ) e a longitude por *lambda* (λ).

Um *paralelo* é o círculo imaginário que une pontos de mesma latitude, formado pela intersecção de um plano paralelo ao Equador com a superfície terrestre. Um *meridiano* é um círculo imaginário perpendicular ao Equador e que passa pelos pólos da Terra. Por convenção internacional, determinou-se que o meridiano que passa pelo observatório de Greenwich, na Inglaterra, seja usado como referência origem das coordenadas geográficas, sendo chamado de meridiano origem, meridiano fundamental ou primeiro meridiano (*prime meridian*).

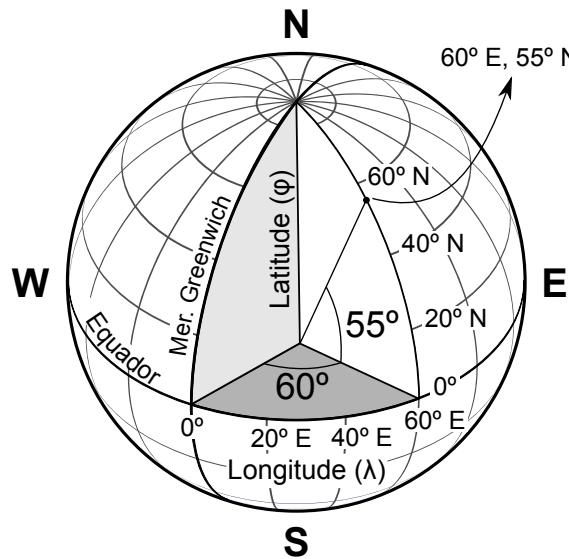


Figura 4.7. Sistema de coordenadas geográficas (latitude-longitude).

Quando falamos de latitude, normalmente estamos nos referindo à *latitude geodética* (ou *latitude geográfica*), que é definida como o ângulo entre o plano equatorial e uma linha normal ao elipsóide de referência, que passe pelo ponto de interesse. O ângulo entre o plano equatorial e uma linha que ligue o ponto de interesse ao centro da Terra é chamado de *latitude geocêntrica*, representada pela letra grega *psi* (ψ) (fig. 4.8). A longitude geocêntrica não difere da longitude geodética, uma vez que os planos de mesma latitude (paralelos) podem ser considerados como círculos perfeitos, ao contrário dos planos formados pelos meridianos.

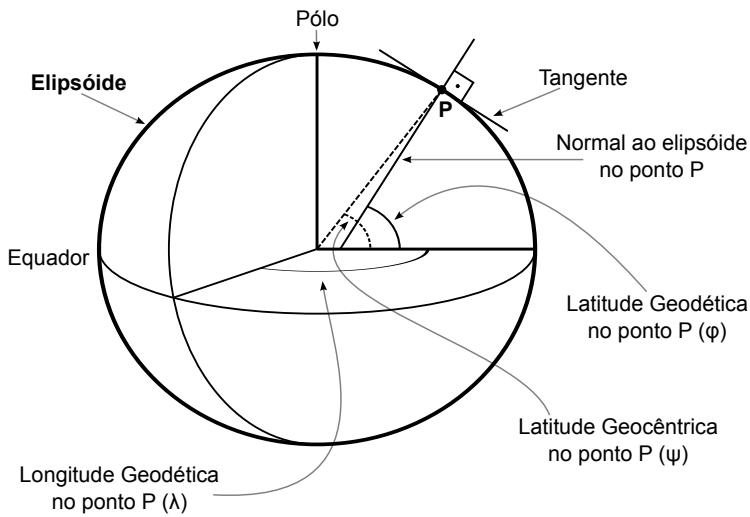


Figura 4.8. Longitude, latitude geodética e latitude geocêntrica.

Considerando os parâmetros do elipsóide de referência, podemos calcular o comprimento do arco equivalente a 1° de latitude ou longitude. Se a Terra fosse uma esfera perfeita, a distância entre dois paralelos (o arco de meridiano) seria sempre a mesma, mas como aproximamos sua forma de um elipsóide, a curvatura da superfície varia mais rapidamente próximo do Equador do que nos pólos, o que faz com que o grau de latitude próximo ao Equador seja ligeiramente menor do que próximo dos pólos. O comprimento do grau de longitude varia de acordo com o cosseno da latitude (tabela 6)⁶, de aproximadamente 111,32 km no Equador até zero nos pólos (fig. 4.9).

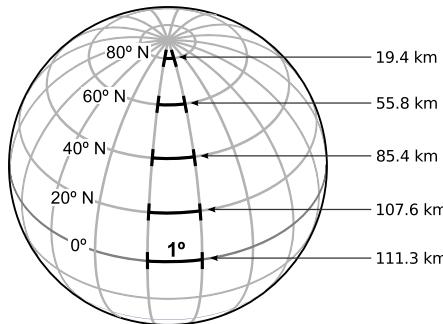


Figura 4.9. Variação do comprimento do grau de longitude com a latitude.

latitude	arco de 1° de latitude (km)	arco de 1° de longitude (km)
0°	110.574 km	111.319 km
10°	110.608 km	109.639 km
20°	110.704 km	104.647 km
30°	110.852 km	96.486 km
40°	111.035 km	85.394 km
50°	111.229 km	71.696 km
60°	111.412 km	55.800 km
70°	111.562 km	30.187 km
80°	111.660 km	19.393 km
90°	111.694 km	0.000 km

Tabela 6. Variação dos comprimentos de graus de latitude e longitude

A distância mais curta entre dois pontos na superfície de uma esfera (ou elipsóide) é dada por um *círculo máximo*, formado pela intersecção entre a superfície da esfera e um plano que passa pelo centro da esfera. Todos os meridianos são círculos máximos, assim como o Equador. Um *círculo mínimo* é definido por um plano que intersecta a esfera mas não passa pelo seu centro. Todos os paralelos são círculos mínimos, exceto quando a latitude for igual a 90°, situação em que um plano perpendicular ao Equador não irá cortar a esfera, mas será tangente à ela. Uma linha que une pontos de mesma direção, ou seja, que cruza os meridianos com o mesmo ângulo é chamada de linha de rumo ou *loxodrómia*.

⁶Os valores da tabela 6 foram calculados no sítio *Length of degree calculator - National Geospatial-Intelligence Agency* – <http://www.nga.mil/MSISiteContent/StaticFiles/Calculators/degree.html>

Conversão entre graus-minutos-segundos e graus decimais

As coordenadas esféricas podem ser especificadas no sistema *sexagesimal*, em graus, minutos e segundos (cada grau é dividido em 60 minutos e cada minuto em 60 segundos) mais uma letra para indicar o quadrante como em $45^{\circ}31'57''\text{W}$, ou em *graus decimais* (valores positivos para Norte e Leste e negativos para Sul e Oeste) como em -45.5325° .

A conversão entre valores no sistema sexagesimal (graus-minutos-segundos) para o sistema decimal (graus decimais) é simples. Basta lembrar que em um grau há 60 minutos e que em cada minuto há 60 segundos (ou seja, em um grau há 3600 segundos). Assim, dividimos os minutos e os segundos por suas frações de grau e somamos os valores.

Como exemplo, vamos converter $45^{\circ}31'57''\text{W}$ em graus decimais (lembre-se que coordenadas Oeste e Sul recebem valores negativos):

$$45^{\circ} + \frac{31'}{60} + \frac{57''}{3600} = 45 + 0.516666 + 0.015833 = -45.5325^{\circ}$$

Para converter valores em graus decimais para a notação graus-minutos-segundos, são necessários alguns passos a mais. Vamos converter -45.5325° de volta para o sistema sexagesimal:

1. A parte inteira do número passa a ser o valor em graus.

$$-45.5325^{\circ} \rightarrow 45^{\circ}$$

2. Multiplica-se a parte decimal por 60. A parte inteira do resultado passa a ser o valor dos minutos.

$$0.5325 \times 60 = 31.95 \rightarrow 31'$$

3. Multiplica-se a nova parte decimal por 60. O resultado será o valor dos segundos (este valor pode ter uma parte decimal ou ser arredondado).

$$0.95 \times 60 = 57 \rightarrow 57''$$

4. Junta-se os resultados em um valor único.

$$-45.5325^{\circ} = 45^{\circ}31'57''\text{W}$$

4.6. Projeções cartográficas

A projeção é necessária para transferir a forma tridimensional da Terra para o plano, bidimensional (mapas). A fim de evitar, tanto quanto possível, a distorção (inevitável), existem diferentes modelos, que podem ser aplicados dependendo da situação, área geográfica, e até mesmo interesse político-econômico.

Podemos classificar as projeções de acordo com a superfície de projeção e de acordo com as propriedades que são preservadas nos mapas (área, forma, distâncias, etc).

De acordo com a superfície de projeção (fig. 4.10):

- *Azimutal ou Plana*: o plano de projeção é posicionado como tangente à superfície da Terra (do elipsóide). Pode-se imaginar uma fonte de luz no infinito, iluminando através do globo, e projetando as sombras dos meridianos e paralelos no plano do mapa.
- *Cilíndrica*: o plano do mapa é tangente ao Equador, e “enrolado” no globo como um cilindro. Os meridianos e paralelos são projetados no plano e criam uma grade retangular.
- *Cônica*: neste caso, o plano do mapa é “enrolado” como um cone, com uma linha de tangência com a Terra (paralelo padrão - *standard parallel*). Freqüentemente o vértice do cone é alinhado com os dois pólos, os meridianos se dispõem radialmente (com o mesmo ângulo) e os paralelos formam círculos concêntricos.
- A superfície de projeção pode ser orientada perpendicularmente ao eixo da Terra (Normal), paralela (Transversa) ou diagonalmente (Oblíqua).
- A superfície de projeção pode tangenciar a Terra, caso em que há um paralelo padrão nas projeções cônicas, ou pode cortar o globo (ser secante ao elipsóide), caso em que haverão dois paralelos padrão (fig. 4.11).

De acordo com as propriedades preservadas:

- *Direção*: azimutal
- *Forma* (localmente): conformal ou ortomórfica
- *Área*: igual-área ou equivalente ou autállica
- *Distância*: equidistante

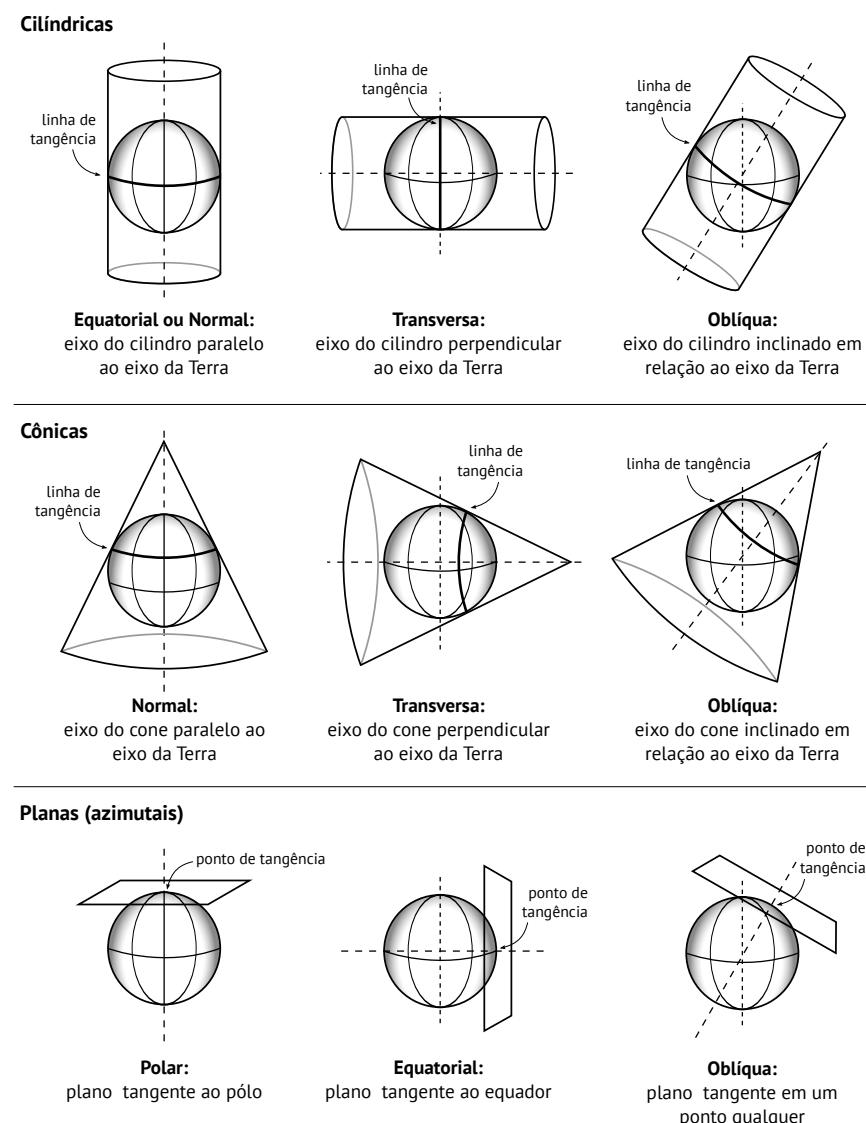


Figura 4.10. Classificação de projeções cartográficas de acordo com a superfície de projeção (IBGE, 1998).

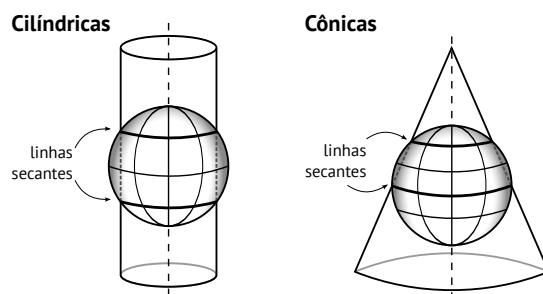


Figura 4.11. Projeções secantes.

A seguir, veremos algumas das principais projeções. Os círculos verdes sobre os mapas são *indicatrizes de Tissot* (cada círculo é uma indicatriz), e representam áreas circulares com o mesmo tamanho dispostos sobre o globo de forma a mostrar as deformações de área e forma (ou ambas) que as projeções acarretam⁷.

4.7. Projeções cilíndricas

Nas projeções cilíndricas, os meridianos são linhas verticais igualmente espaçadas e os paralelos são linhas horizontais. Essas projeções diferenciam-se pelo tipo de deformação ao longo da latitude (direção N-S).

4.7.1 Cilíndrica conformal (Mercator)

Esta é um projeção conformal, portanto preserva-se a forma dos elementos do mapa. Na projeção de Mercator, a escala N-S é igual à escala E-W, o que leva a uma deformação excessiva em altas latitudes.

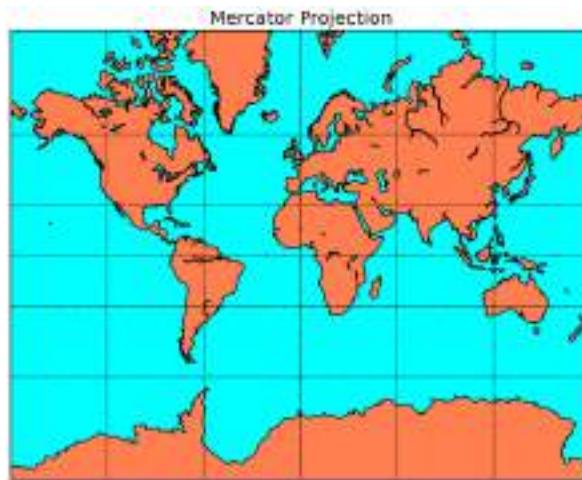


Figura 4.12. Projeção cilíndrica conforme de Mercator.

4.7.2 Cilíndrica equidistante

Também chamada de equiretangular ou *plate carré* (literalmente, placa quadrada), não é nem conformal nem de igual-área. Os meridianos e paralelos são igualmente espaçados e a escala N-S é constante, portanto longitude e latitude são mapeadas diretamente para x e y e o mapa-mundi é um retângulo com proporção 2:1.

⁷As figuras das projeções foram retiradas de <http://matplotlib.github.com/basemap/users/mapsetup.html>

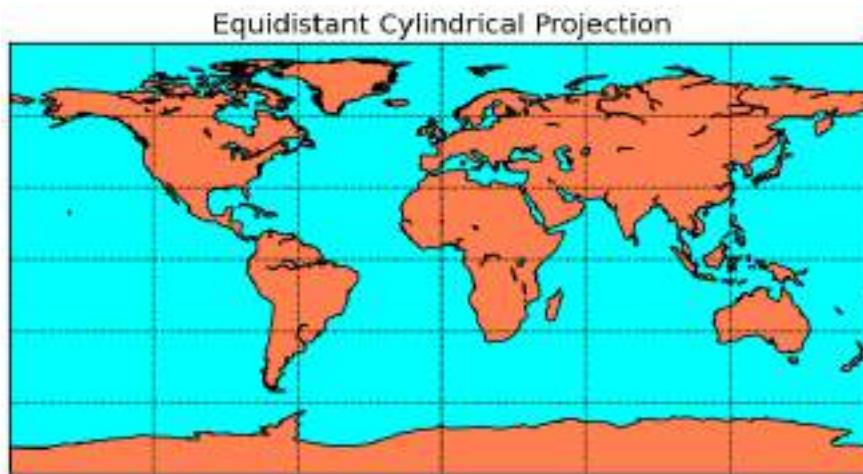


Figura 4.13. Projeção cilíndrica equidistante.

4.8. Projeções cônicas

As projeções cônicas são mais usadas em áreas com extensão maior E-W, fora do Equador.

4.8.1 Cônica conformal

Uma projeção secante com dois paralelos padrão (projeção cônica conforme de Lambert).

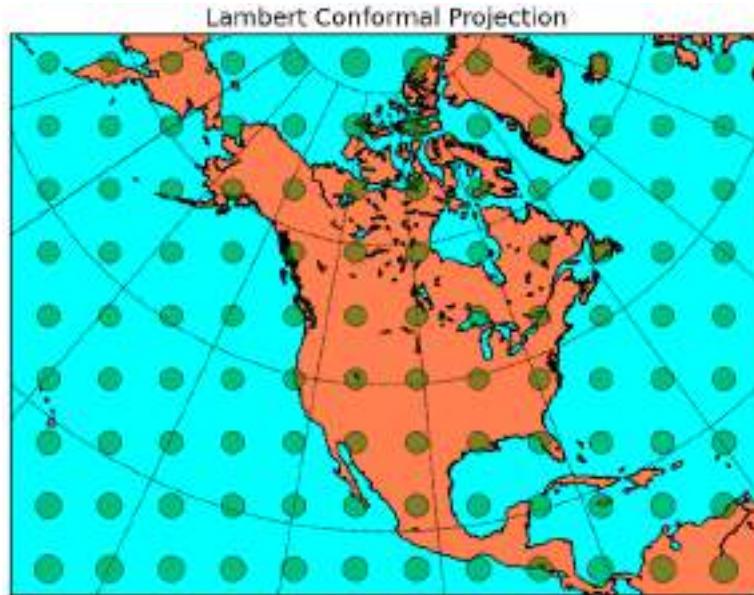


Figura 4.14. Projeção cônica conforme de Lambert.

4.8.2 Cônica igual-área

Projeção secante com dois paralelos padrão. Na projeção de Albers, a distorção entre os paralelos padrão é mínima.

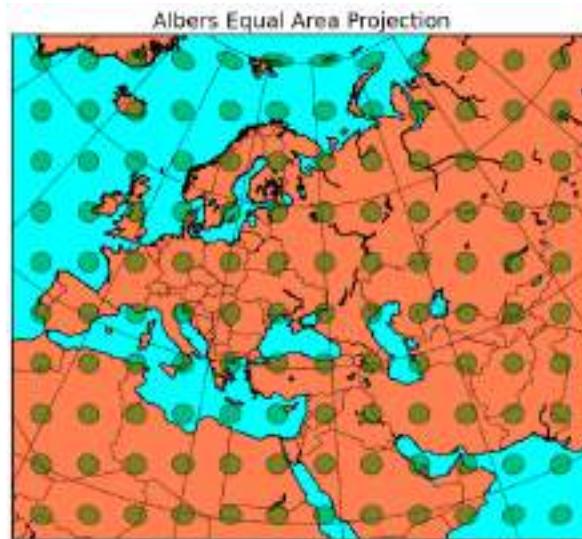


Figura 4.15. Projeção cônica igual-área de Albers.

4.8.3 Cônica equidistante

Nesta projeção, os paralelos são igualmente espaçados. Utilizada para áreas próximas do paralelo padrão.

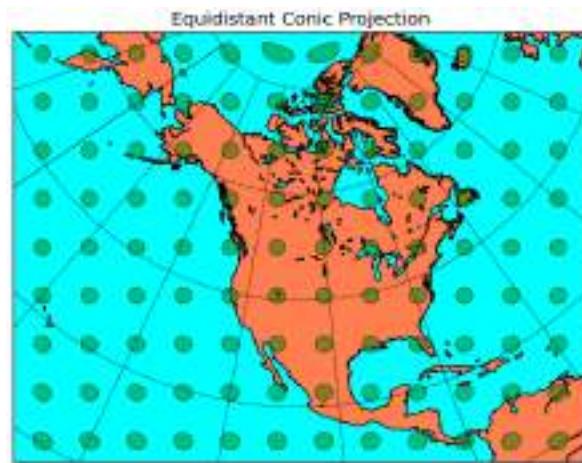


Figura 4.16. Projeção cônica equidistante.

4.8.4 Policônica

Indicada para áreas com grande extensão N-S. Não é conformal ou igual-área. Baseada em múltiplos cones, cada um com um paralelo padrão. Utilizada em mapas da América do Sul e do Brasil.



Figura 4.17. Projeção policônica.

4.9. Projeções azimutais

4.9.1 Azimutal igual-área

Nesta projeção, as direções válidas a partir do ponto central. Toda linha reta que passe pelo centro é um círculo máximo. Em geologia estrutural, é chamada de projeção de Schmidt-Lambert.

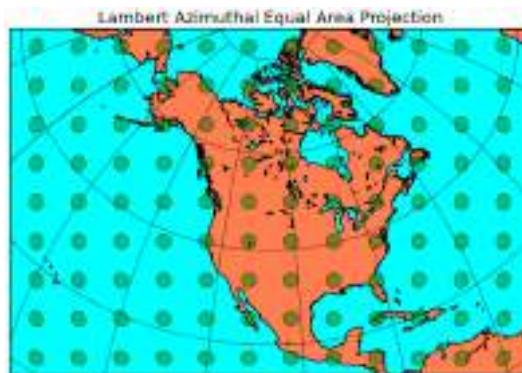


Figura 4.18. Projeção azimutal igual-área de Lambert.

4.9.2 Azimutal equidistante

As distâncias e direções são válidas para qualquer ponto a partir do centro do mapa. Toda linha reta que passe pelo centro é um círculo máximo.

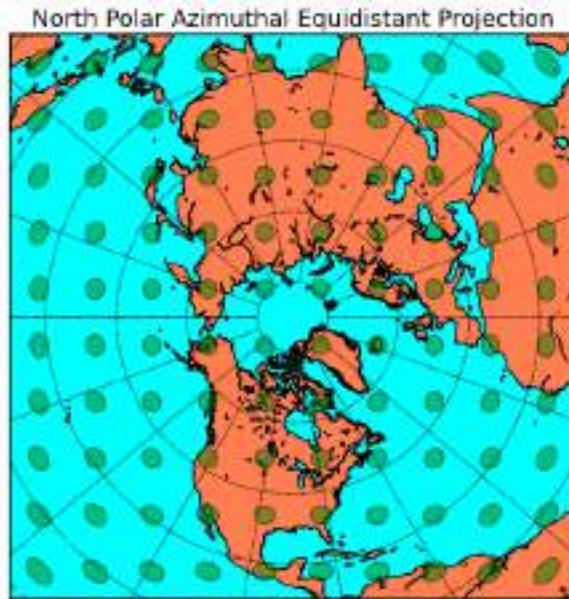


Figura 4.19. Projeção polar azimutal equidistante.

4.9.3 Azimutal ortográfica

Ponto de perspectiva no infinito. Mostra o planeta como um fotografia a partir da Lua.

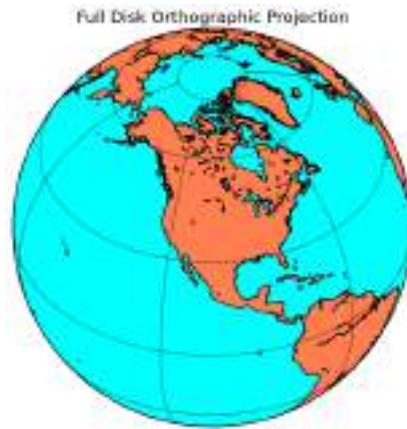


Figura 4.20. Projeção azimutal ortográfica.

4.9.4 Azimutal gnomônica

Ponto de perspectiva no centro do globo. Não é conformal, nem igual-área nem equidistante. Círculos máximos são linhas retas.

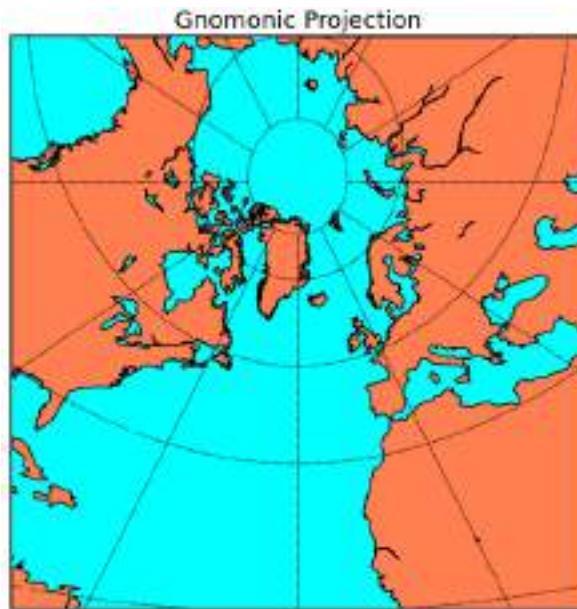


Figura 4.21. Projeção azimutal gnomônica.

4.10. Projeções compromisso

4.10.1 Projeção de Robinson

Não é conformal, nem igual-área, foi usada pela National Geographic Society para seus mapas-mundi.

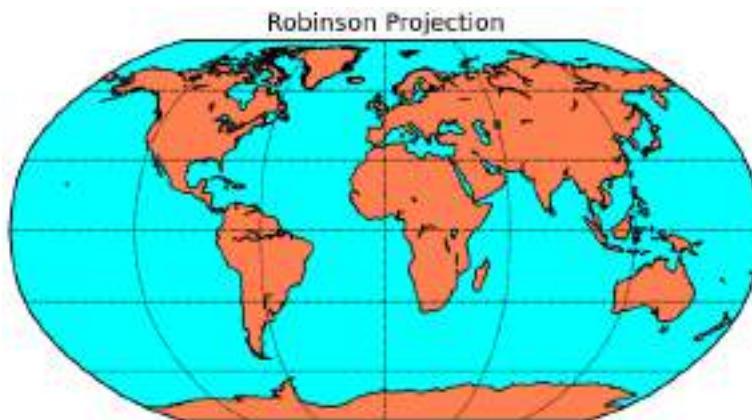


Figura 4.22. Projeção azimutal de Robinson.

4.11. Coordenadas UTM

No sistema UTM (*Universal Transverse Mercator*), utiliza-se a projeção cilíndrica transversa de Mercator. A Terra é dividida em 60 zonas (fusos) com 6° de longitude cada, entre 84° de latitude norte e 80° de latitude sul, com exceção de algumas áreas nas regiões próximas aos pólos (fig. 4.23). Cada zona UTM é mapeada por um cilindro transverso, com linha de tangência no meridiano central da zona.

As coordenadas são indicadas por E (Este) e N (Norte) e têm sua origem no Meridiano central de cada zona e no Equador. Para evitar valores negativos, o meridiano central recebe o valor de 10.000 km e o Equador, 500 km (fig. 4.24).

Os meridianos centrais estão localizados em 3° , 9° , 15° , e assim por diante, para Leste e Oeste. As zonas são ainda divididas em faixas de sul para norte, indicadas por letras (fig. 4.23). As faixas marcadas com letras de A até M estão no hemisfério Sul, e de N até Z no hemisfério Norte.

Note que a notação completa das coordenadas UTM inclui a letra da faixa, como em **23K 500.000E / 7.500.000N**. Na figura 4.24 fica claro que o mesmo valor numérico de coordenadas ocorre nos dois hemisférios⁸, portanto indicar a letra da faixa é fundamental. Um ponto importante a lembrar é que os programas de SIG usualmente diferenciam as zonas UTM apenas por Norte ou Sul, como em **23N** e **23S**. Isso funciona para os programas, mas devemos ficar atentos para não cometer o erro de indicar uma coordenada “23S” para uma localidade no hemisfério Sul, já que a zona 23S está no hemisfério Norte.

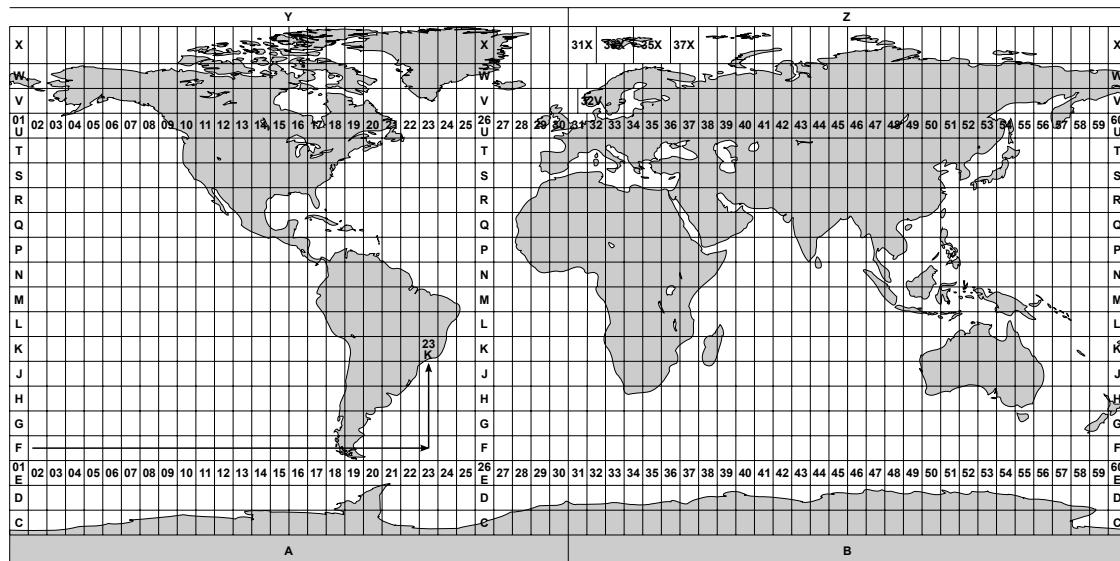


Figura 4.23. Divisão das zonas UTM.

⁸Na verdade, o mesmo valor numérico ocorre nas 60 zonas, nos dois hemisférios, ou seja, 120 vezes.

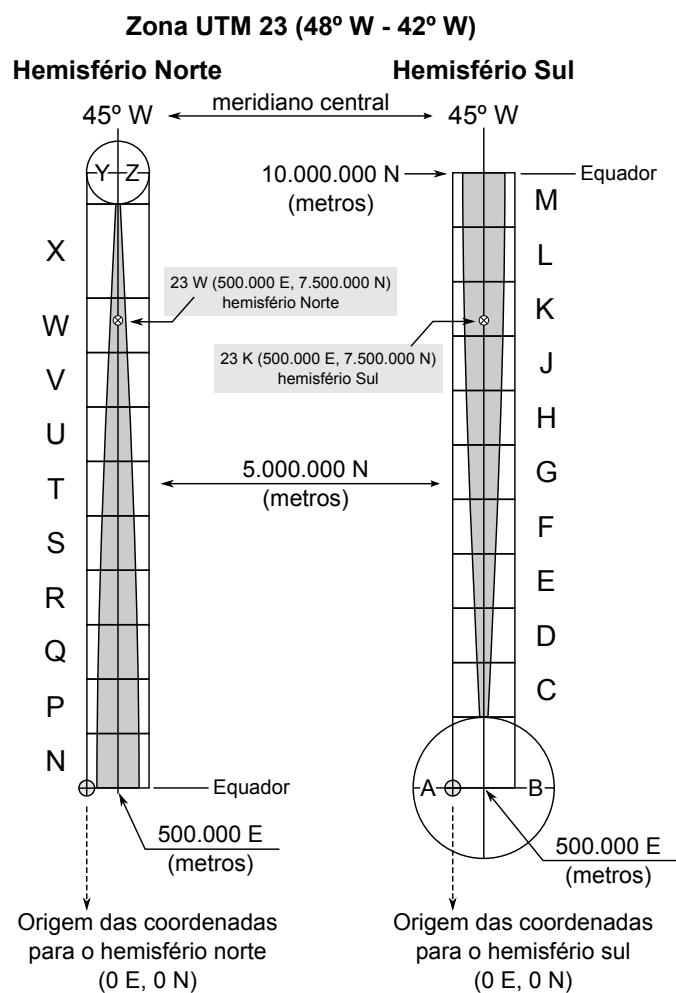


Figura 4.24. Origem das coordenadas nas zonas UTM.

4.12. Carta Internacional do Mundo ao Milionésimo

Em 1891, o alemão Albrecht Penck propôs uma divisão do mundo em cartas na escala 1:1.000.000. Essa divisão é feita em fusos de 6° de longitude, numerados de 1 a 60 a partir do antimeridiano de Greenwich para Leste. Cada fuso é dividido em faixas de 4° de latitude, marcadas, a partir do Equador, pelas letras A, B, C, D, E, F,... mais a notação do hemisfério (N ou S). A figura 4.25 mostra a articulação da Carta Internacional ao Milionésimo em território brasileiro.



Figura 4.25. Articulação da Carta Internacional ao Milionésimo em território brasileiro.

4.13. Sistema cartográfico nacional

4.14. Padrão de exatidão cartográfica

O decreto federal nº 89.817 de 20 de junho de 1984 (Anexo ??) estabelece as Normas Técnicas da Cartografia Nacional. Entre outras deliberações, determina-se que os produtos cartográficos devem obedecer ao Padrão de Exatidão Cartográfica (PEC), onde 90% dos pontos no terreno não podem ter erro de posicionamento (planimétrico e altimétrico) maior do que o estabelecido pelo PEC, e que os mapas são classificados de acordo com os seguintes critérios:

- Classe A

Erro planimétrico: 0,5 mm na escala da carta

Erro altimétrico: metade da equidistância entre as curvas de nível

- Classe B

Erro planimétrico: 0,8 mm na escala da carta

Erro altimétrico: três quintos da equidistância entre as curvas de nível

- Classe C

Erro planimétrico: 1,0 mm na escala da carta

Erro altimétrico: três quartos da equidistância entre as curvas de nível

4.15. Mapeamento sistemático

O mapeamento sistemático topográfico do Brasil é de responsabilidade do IBGE e compreende as seguintes escalas: 1:1.000.000, 1:500.000, 1:250.000, 1:100.000, 1:50.000 e 1:25.000. Mapeamentos em escala maior são considerados cadastrais e de responsabilidade dos estados e municípios. Suas escalas normais variam de 1:10.000 até 1:2.000.

Os mapas com escala de 1:250.000 a 1:25.000 utilizam coordenadas UTM, enquanto que os mapas em 1:1.000.000 e 1:500.000 estão em coordenadas geográficas, na projeção Cônica Conforme de Lambert.

A articulação dos mapas segue uma nomenclatura que indica tanto a escala do mapa quanto sua localização dentro da carta ao milionésimo. O índice é baseado em divisões sucessivas dos mapas, onde letras ou números representam cada etapa. O mapa inicial na escala 1:1.000.000 é dividido em 4 mapas 1:500.000 (V, X, Y, Z). Cada mapa 1:500.000 é dividido em 4 mapas 1:250.000 (A, B, C, D) e cada um desses em 6 mapas 1:100.000 (I, II, III, IV, V, VI). Os mapas 1:100.000 são divididos em 4 mapas 1:50.000 (1, 2, 3, 4) e finalmente cada mapa 1:50.000 é dividido em 4 mapas 1:25.000 (NO, NE, SE, SO).

As figuras 4.26 e 4.27 ilustram o processo. Veja que a partir do índice de nomenclatura de um mapa é possível determinar sua escala:

1 : 1.000.000 SF-23 – Rio de Janeiro
1 : 500.000 SF-23-X
1 : 250.000 SF-23-X-B – Ponte Nova
1 : 100.000 SF-23-X-B-I
1 : 50.000 SF-23-X-B-I-1 – Catas Altas
1 : 25.000 SF-23-X-B-I-1-SO

Outro modelo utilizado para indexar os mapas no Brasil é o Mapa Índice (MI), que atribui uma numeração sequencial aos mapas de acordo com a escala. Na escala 1:50.000, acrescenta-se ao código uma numeração (1, 2, 3, 4) e para 1:25.000 acrescenta-se um indicador (NO, NE, SE, SO), tal como no esquema anterior:

- 1 : 1.000.000 numeração de 1 a 46
 1 : 250.000 numeração de 1 a 550
 1 : 100.000 numeração de 1 a 3036 (p.ex., MI 2574)
 1 : 50.000 código MI + numeração 1, 2, 3, 4
 1 : 25.000 código MI 1:50.000 + indicador NO, NE, SE, SO

A tabela 7 apresenta ambos os sistemas de indexação:

Escala	Índice de nomenclatura	Índice MI
1 : 1.000.000	SF-23 – Rio de Janeiro	MI-39
1 : 500.000	SF-23-X	—
1 : 250.000	SF-23-X-B – Ponte Nova	MI-0466
1 : 100.000	SF-23-X-B-I	MI-2574
1 : 50.000	SF-23-X-B-I-1 – Catas Altas	MI-2574-1
1 : 25.000	SF-23-X-B-I-1-SO	MI-2574-1-SO

Tabela 7. Nomenclaturas de mapas no Sistema Cartográfico Nacional

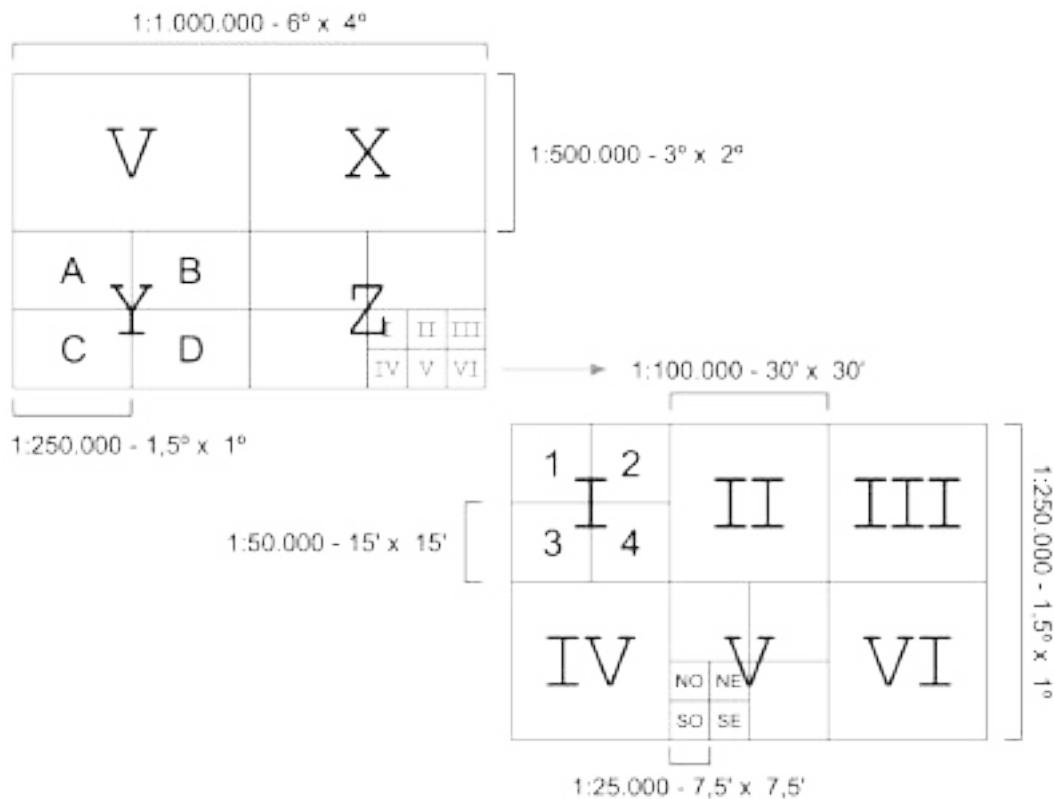


Figura 4.26. Articulação do sistema cartográfico nacional (visão condensada).

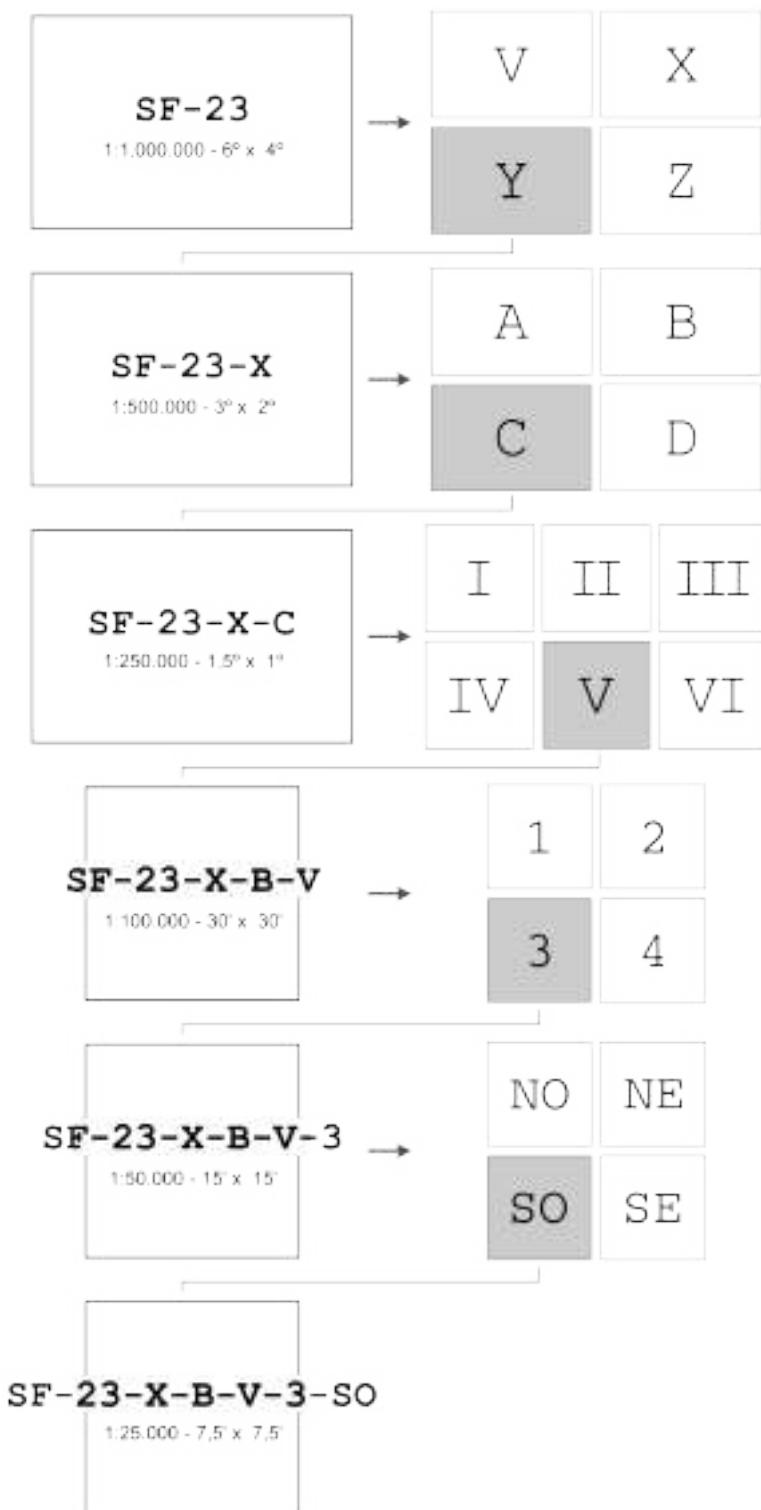


Figura 4.27. Articulação do sistema cartográfico nacional (visão expandida).

5. Bem-vindo(a) ao GRASS-GIS!

5.1. Criando uma Location e um mapset

O primeiro passo ao trabalhar com GRASS é a criação de uma **Location** e de ao menos um **mapset**. Ao rodar o programa pela primeira vez, uma janela de terminal (também chamada de console) como a da figura 5.1 é exibida.

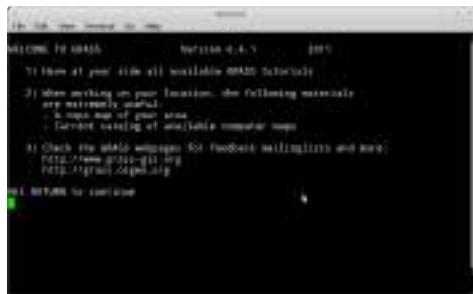


Figura 5.1. Tela de terminal com a mensagem de boas-vindas ao GRASS-GIS.

Na figura 5.2 temos o campo superior central (*GIS Data Directory*), com o caminho para o diretório (pasta) **GISBASE**, onde são armazenados os dados dos projetos. Na primeira vez que rodamos o GRASS, esse diretório não está definido, e recebemos a mensagem em vermelho com um alerta. Das próximas vezes, essa mensagem não irá aparecer. No campo inferior da esquerda estão listadas as **Locations**; cada uma é um subdiretório de **GISBASE**. No campo inferior central, são listados os **mapsets**; novamente, cada um é um subdiretório da sua **Location**. No campo da direita podemos criar novas **Locations** e **mapsets** ou renomear/deletar **mapsets**.



Figura 5.2. Janela inicial do GRASS.

Para criar uma Location, clique no botão **Location wizard**. A primeira etapa é definir o caminho para o diretório GISBASE (**GIS Data Directory**, comumente usamos `grassdata`), e o nome da Location (fig. 5.3). O título é opcional.



Figura 5.3. Definição do diretório GISBASE e do nome da Location.

Na tela da figura 5.4 decidimos como criar a Location. Temos várias opções: selecionar parâmetros de listagens; utilizar códigos pré-definidos pela *European Petroleum Survey Group* (EPSG) para diversas combinações de *datums* e projeções cartográficas; ler os parâmetros a partir de arquivos georreferenciados (como os GeoTIFFs), de arquivos WKT (*Well-Known Text*) ou PRJ (arquivos com informações de projeção que acompanham dados vetoriais em formato *shapefile*); definir projeções e datums manualmente ou ainda criar sistemas arbitrários XY. Selecione a primeira opção (EPSG codes).

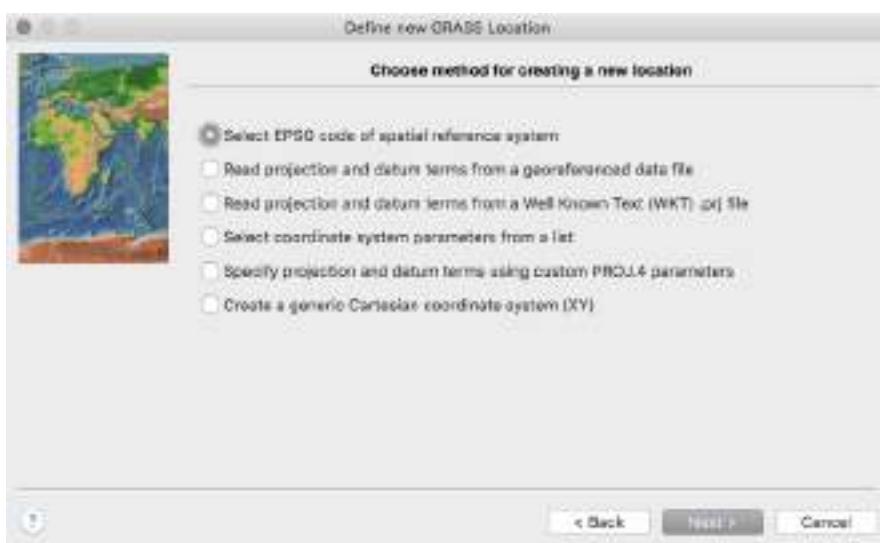


Figura 5.4. Defina como será o método de criação da Location.

Ao determinar os parâmetros da Location pelos códigos EPSG, já definimos a projeção (com o sistema de coordenadas) e o *datum* (fig. 5.5). No nosso exemplo, selecionamos o código 32722, que corresponde à coordenadas UTM, na zona 22, hemisfério Sul, com *datum* WGS-84.

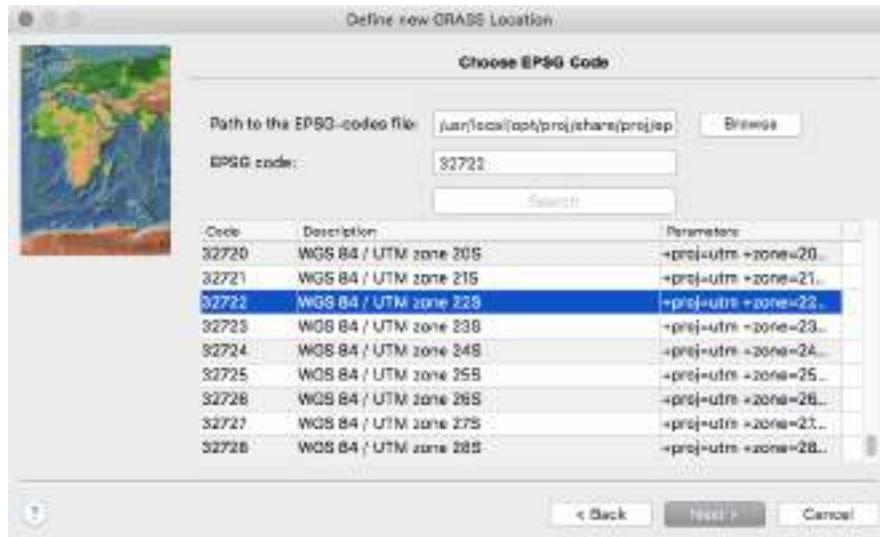


Figura 5.5. Defina a projeção e o sistema de coordenadas.

A seguir definimos o parâmetro de transformação de coordenadas de acordo com o *datum* escolhido e a região geográfica (fig. 5.6). Selecione a opção 1.



Figura 5.6. Escolha do *datum* WGS-84.

Agora uma tela com um resumo das suas escolhas é exibida (fig. 5.7). Se tudo estiver certo, clique em **Finish**.

Definida a Location, você será perguntado se quer definir os limites e a resolução espacial da **region** padrão (fig. 5.8). Clique em **Yes**.

Na janela da figura 5.9, os campos ao redor do mapa indicam os limites geográficos da **region** padrão. no nosso exemplo, usaremos **North:7405000**, **South:7160000**, **West:640000**, **East:885000**. A resolução N-S e E-W será de 30m, que é a resolução dos dados SRTM que utilizaremos.



Figura 5.7. Confira se está tudo certo.



Figura 5.8. Quase lá.



Figura 5.9. Defina os limites e a resolução da region padrão.

Esses valores são metros, já que nossa Location está em UTM. Valores em graus podem estar no formato <graus:minutos:segundos N/S/E/W> como em <19:36:00S> ou em graus decimais (com valores negativos para os hemisférios Sul e Oeste), como em <-19.6>, e a resolução como em <0:00:03> ou <0.0008333>.

Muito bem, agora a Location está pronta. Clique em **New** na parte direita da janela e defina seu mapset da maneira que achar melhor (fig. 5.10).



Figura 5.10. Defina o nome do seu mapset.

Agora sim. Criado o mapset, clique em **Start GRASS!** Temos agora três janelas (fig. 5.11): uma de terminal (em baixo, à esquerda), onde podemos digitar comandos; uma janela chamada de **GIS Manager** (em cima, à esquerda), onde teremos os arquivos abertos para visualização e os menus de comandos; e uma janela de **Map Display** (em cima, à direita), onde os dados são exibidos. Diversos *Map Displays* podem ser abertos ao mesmo tempo. Uma descrição mais detalhada das funcionalidades do *GIS Manager* e do *Map Display* serão discutidas no próximo capítulo.

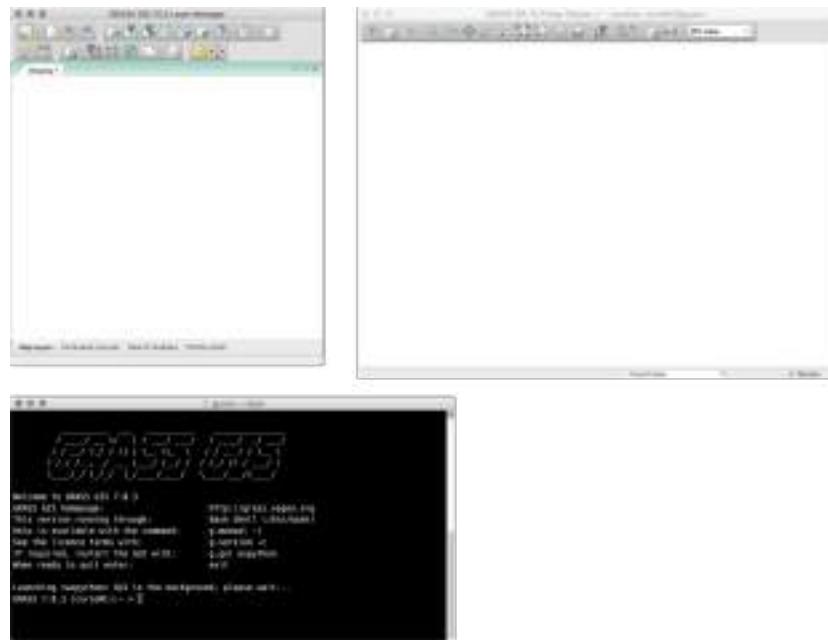


Figura 5.11. Interface gráfica do GRASS-GIS 7

5.2. Apresentando o *Layer Manager* e o *Map Display*

Neste capítulo veremos brevemente as funcionalidades do *GIS Manager* e do *Map Display*. No *Layer Manager* (fig. 5.12) encontramos todos os comandos separados em menus, e alguns dos mais utilizados estão agrupados em uma barra de ferramentas (fig. 5.13). Abaixo da barra de ferramentas temos a área onde são organizados os diversos *layers*, vetoriais e *raster*, além de composições coloridas (RBG ou IHS), gráficos e elementos cartográficos, como barras de escala e seta de norte. Note que o empilhamento vertical dos *layers* no *Layer Manager* reflete a ordem em que serão exibidos no *Map Display*.

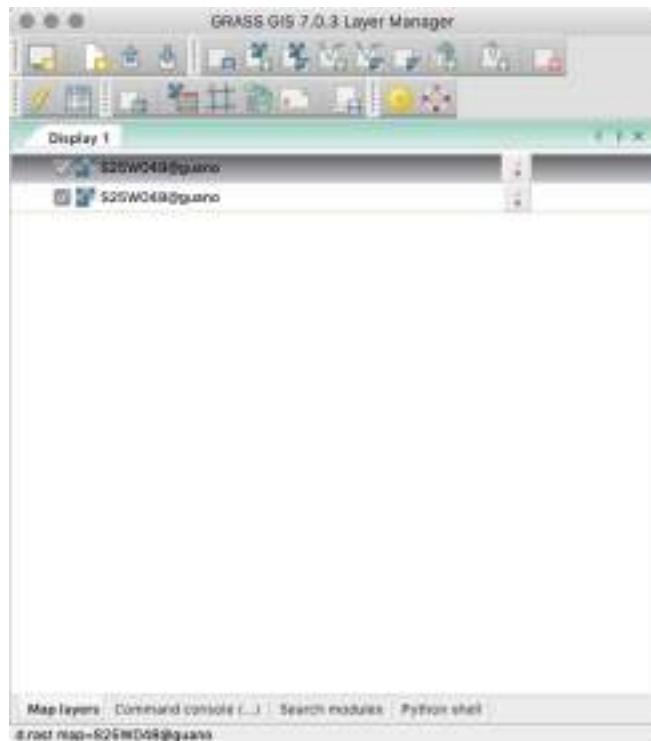


Figura 5.12. Janela do *Layer Manager*.

Na porção inferior da janela do *Layer Manager*, temos quatro abas: **Map layers**, **Command console**, **Search module** e **Python Shell**. Na aba *Map layers* ficarão os layers carregados e que podem ser exibidos no *Map Display*. A aba *Command console* é utilizada para exibir informações de saída (*output*) de alguns comandos e também possui um *console*, ou *terminal*, onde podemos digitar comandos (mais utilizado em ambiente Windows, onde a janela de terminal que vimos na figura 5.11 não é muito utilizada). Na aba *Search module*, podemos buscar módulos (comandos) a partir de parte do nome ou da descrição de sua funcionalidade. Na aba *Python Shell* temos um terminal de linguagem Python, que podemos usar para trabalhar os dados dos GRASS com auxílio de linguagem de programação.

Na figura 5.13, temos barra de ferramentas do *Layer Manager*, com a indicação das funcionalidades e na figura 5.14, a barra de ferramentas do *Display Manager*.

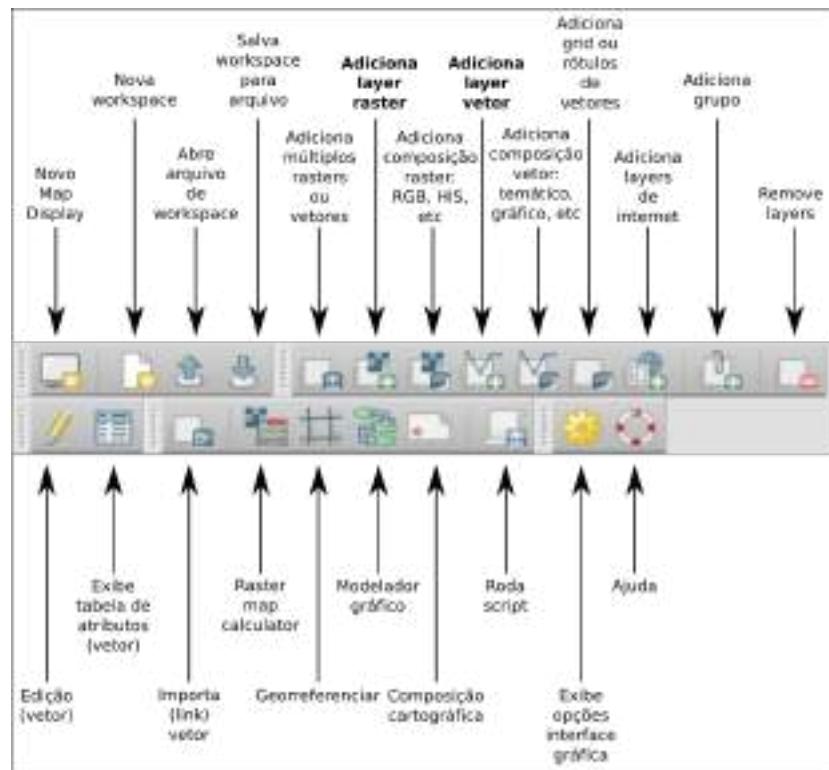


Figura 5.13. Barra de ferramentas do *Layer Manager*.

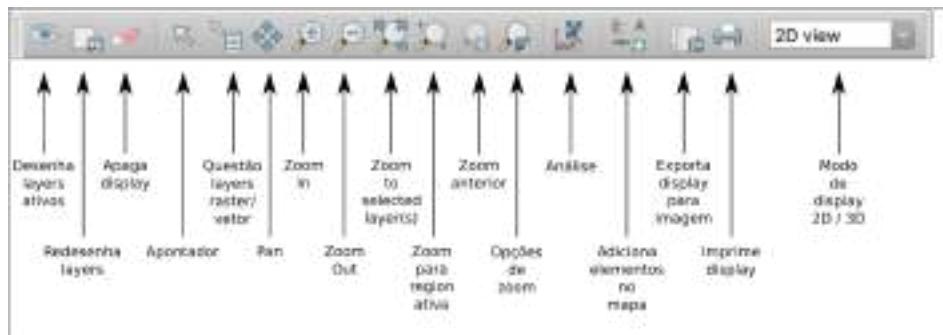


Figura 5.14. Barra de ferramentas do *Map Display*.

6. Dados Raster

Vamos começar a utilizar o GRASS com dados tipo raster. Isso nos permitirá aprender como baixar dados de elevação SRTM, como trabalhar com esses dados, e ainda, a trabalhar com o conceito de `region`, algho exclusivo do GRASS e extremamente poderoso.

6.1. Modelos de Elevação SRTM

A missão *Shuttle Radar Topography Mission* (SRTM – Farr & Kobrick, 2000; van Zyl, 2001; Rabus *et al.*, 2003; Farr *et al.*, 2007) foi realizada pela agência espacial norte-americana (*National Aeronautics and Space Administration* – NASA), a *National Geospatial-Intellingence Agency* (NGA), o Departamento de Defesa dos Estados Unidos (DoD) e as agências espaciais alemã (*Deustches Zentrum für Luft- und Raumfahrt* – DLR) e italiana (*Agenzia Spaziale Italiana* – ASI) em fevereiro de 2000, para mapear o relevo da área continental da Terra com interferometria de radar de abertura sintética (InSAR), entre 60° de latitude norte e 54° de latitude sul, o que corresponde à aproximadamente 80% das áreas emersas do planeta.

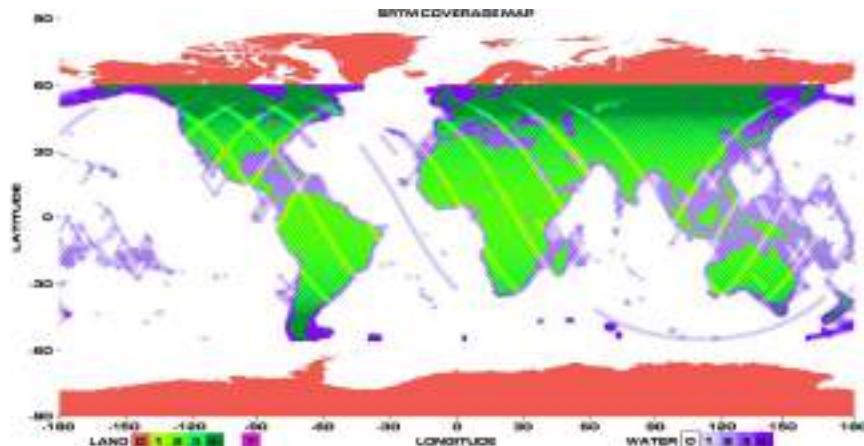


Figura 6.1. Mapa da cobertura dos dados SRTM (Farr & Kobrick, 2000).

Os princípios da InSAR foram desenvolvidos no início dos anos 1980 (Zebker & Goldstein, 1986). Assim como a estereoscopia óptica se baseia na paralaxe causada pela diferença de posição do observador com relação ao ponto de interesse (Lillesand & Kiefer, 1987), a altitude de um ponto na superfície pode ser determinada a partir da diferença de fase entre reflexões de um mesmo sinal de radar captado por dois receptores distantes um do outro (Rabus *et al.*, 2003).

No caso da missão SRTM, os canais principais de transmissão e recepção para as bandas C e X foram posicionados na área de carga do ônibus espacial *Endeavour*, enquanto que as antenas secundárias (sómente recepção) foram colocadas na extremidade de um mastro retrátil de 60m de comprimento. Para uma descrição detalhada de interferometria de radar de visada lateral, consultar, por exemplo, Bamler & Hartl (1998).

Deve-se levar em conta que o resultado das técnicas empregadas na missão SRTM é a geração de Modelos Digitais de *Elevação* (MDEs), pois os sinais de radar são refletidos, por exemplo, pelo dossel das árvores em áreas densamente florestadas e não pelo terreno subjacente (fig. 6.2). Os termos Modelo Digital de *Terreno* (MDT) e Modelo Numérico de *Terreno* (MNT) devem ser reservados para casos onde o modelo é produzido a partir de valores de altitude do *nível do terreno*, obtidos, por exemplo, em mapas topográficos (curvas de nível), levantamentos por GPS ou por altimetria a laser (LiDAR).

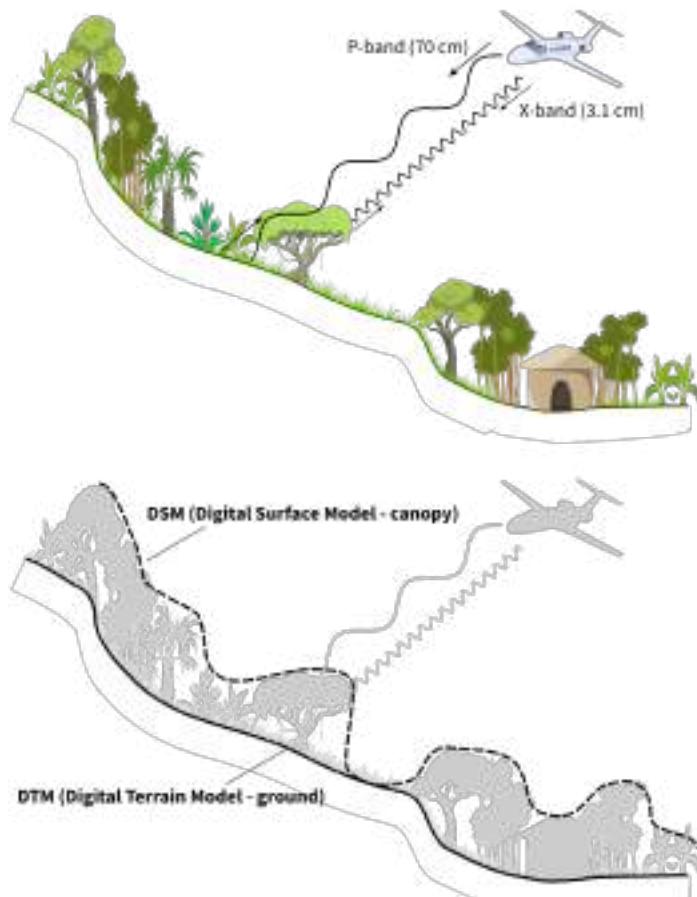


Figura 6.2. Comparação entre a representação topográfica de Modelos Digitais de Terreno (MDTs) e Modelos Digitais de Elevação (MDEs).

6.1.1 Versões dos dados SRTM

NASA SRTM V1 – A primeira versão dos dados SRTM consiste dos MDEs originais produzidos com os dados da missão STS-99 e foi lançada em 2003. São dados brutos, não editados, que contêm valores espúrios em áreas de baixo retorno dos sinais de radar, como em corpos d'água. Podem ser acessados em diretamente no servidor do USGS ⁹.

⁹SRTM v1 – <http://dds.cr.usgs.gov/srtm/version1/>

NASA SRTM V2 e V2.1 – A segunda versão dos dados resultou de um esforço de edição considerável por parte da NGA para remoção de valores espúrios (*single pixel errors*) e definição de linhas de costa e de corpos d'água, apesar de ainda conter áreas de vazios. Esta versão traz arquivos vetoriais (em formato *shapefile*) da máscara de linhas de costa utilizada na edição (*SRTM Water Body Data - SWBD*). A versão 2 foi gerada pelo valor do pixel central da máscara 3x3 (ver fig. 6.4) e continha alguns artefatos, como um bandamento acima de 50° de latitude.

A versão 2.1 (também conhecida como “*finished version*”) representa um reprocessamento dos dados originais com um filtro de média 3x3 e foi lançada em 2005. O produto final possui precisão vertical global de $\pm 16\text{m}$ e horizontal de $\pm 20\text{m}$ (Rabus *et al.*, 2003).

Para a América do Sul, a precisão vertical é de 6,2m e a horizontal de 9,0m (Rodriguez *et al.*, 2006). Esta versão pode ser acessada no servidor do USGS¹⁰, onde os diretórios SRTM1, SRTM3 e SRTM30 representam dados com resolução espacial de 1 segundo, 3 segundos e 30 segundos de arco, respectivamente.

Os dados para o território brasileiro processados pela Embrapa, com resolução espacial de 3 segundos de arco, (Miranda, 2005) estão disponíveis segundo a articulação das folhas topográficas de escala 1:250.000¹¹.

NASA SRTM V3 – A terceira versão dos dados SRTM (também chamada de “*SRTM Plus*”) foi liberada em 20 de novembro de 2013 pelo *Land Processes Distributed Active Archive Center* (LP DAAC). O objetivo principal dessa versão foi a eliminação completa dos vazios (*voids*), que foram preenchidos principalmente com dados ASTER GDEM2 e também com dados USGS GMTED2010 ou USGS NED¹² (*National Elevation Dataset*) para áreas dos Estados Unidos (exceto Alasca) e do norte do México.

O processamento dos dados SRTM versão 3.0 foi realizado dentro do programa NASA MEaSUREs¹³ (*Making Earth System Data Records for Use in Research Environments*). Os dados podem ser acessados pelo serviço *Reverb*¹⁴, pelo *LPDAAC Data Pool*¹⁵ ou em *Open-Topography.org*¹⁶.

Em setembro de 2014, foi anunciada a liberação ao público dos dados globais (versão 3.0) com resolução de 1 segundo de arco (NGA, 2014; JPL, 2014). Os dados podem ser acessados diretamente no servidor do USGS¹⁷.

¹⁰SRTM v2.1 – http://dds.cr.usgs.gov/srtm/version2_1/

¹¹Embrapa – <http://www.relevobr.cnpm.embrapa.br>

¹²NED – <http://ned.usgs.gov>

¹³MEaSUREs – <https://earthdata.nasa.gov/our-community/community-data-system-programs/measures-projects>

¹⁴Reverb – <https://reverb.echo.nasa.gov/reverb>

¹⁵LPDAAC – https://lpdaac.usgs.gov/data_access/data_pool/

¹⁶OpenTopography – <http://www.opentopography.org/id/OTSRTM.042013.4326.1>

¹⁷USGS (SRTM V3 01”) – <http://e4ftl01.cr.usgs.gov/SRTM/SRTMGL1.003/2000.02.11/>

CGIAR-CSI SRTM V4.1 – Esta versão é produzida pelo *Consortium for Spatial Information* (CSI) do *Consultative Group for International Agricultural Research* (CGIAR). Os dados foram processados de modo a fornecer uma cobertura topográfica contínua e sem vazios para todo o mundo (Jarvis *et al.*, 2008). O processamento seguiu os métodos propostos por (Reuter *et al.*, 2007) e os dados são distribuídos em arquivos com 5°x5° com resolução espacial de 3 segundos de arco e podem ser acessados no servidor do CGIAR-CSI¹⁸.

DLR SRTM X-SAR – Desde dezembro de 2010, a agência espacial alemã (DLR) disponibiliza para acesso gratuito os dados SRTM X-SAR (Eineder *et al.*, 2001) com resolução de 1 segundo de arco. Esses dados possuem cobertura bastante inferior à dos dados da banda C, dada a largura menor da faixa de varredura do sensor da banda X (fig. 6.3) ¹⁹.

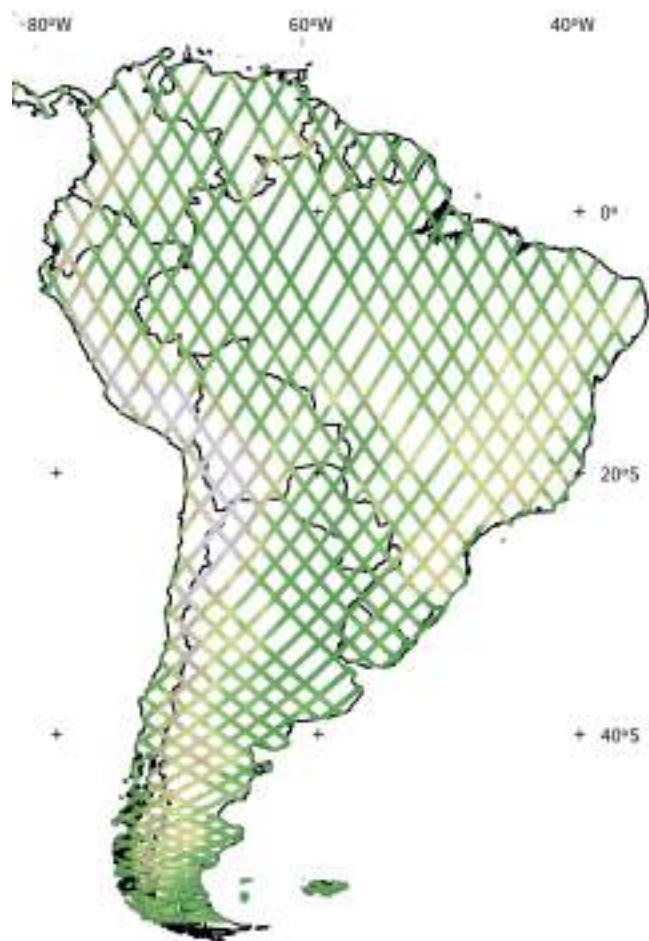


Figura 6.3. Mapa da cobertura dos dados SRTM X-SAR para a América do Sul.

¹⁸CGIAR-CSI – <http://srtm.csi.cgiar.org/SELECTION/inputCoord.asp>

¹⁹SRTM-X – http://www.dlr.de/eoc/en/Portaldata/60/Resources/dokumente/7_sat_miss/srtm_products_en.pdf

6.1.2 Resolução espacial e reamostragem de dados SRTM 03"

Uma característica importante a ser observada é a metodologia empregada na reamostragem dos dados originais para 3 segundos de arco (versões 1, 2 e 2.1). Os dados distribuídos pelo USGS foram gerados a partir da *média* dos valores originais, em uma janela de 3x3 pixels, o que resulta em um modelo suavizado com relação ao original. Os dados processados pela NGA foram gerados a partir do *valor do pixel central* da janela 3x3 e portanto possuem informações dos dados originais (Fig. 6.4).

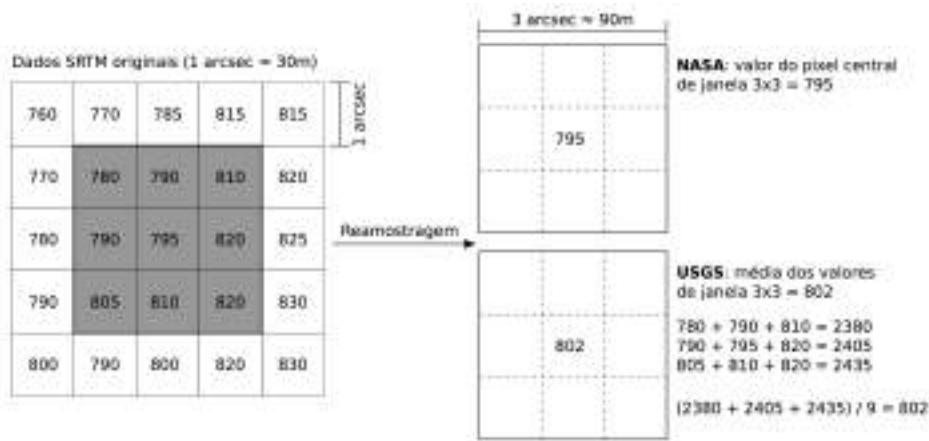


Figura 6.4. Métodos de reamostragem dos dados SRTM utilizados pela NASA e pelo USGS, para geração dos modelos com resolução de 3 segundos de arco.

Na versão 3.0, processada pela NASA, essa metodologia foi mantida. Os dados gerados pela média da janela 3x3 são referenciados como **SRTMGL3**, e os dados que trazem o valor do pixel central da janela 3x3 como **SRTMGL3S** (*sub-sampled*).

Pode-se argumentar que o processamento pela média dos valores produz um resultado superior ao do valor do pixel central, pois reduz o ‘ruído’ de alta frequência presente em dados de elevação derivados de radar. A resolução espacial real dos dados de 1 segundo de arco é estimada entre 50-80m. Os modelos de 3 segundo de arco derivados pelo pixel central (*sub-sampled*) têm resolução espacial da ordem de 100 m, mas com maior nível de ruído do que os derivados pela média, que têm resolução levemente menor, cerca de 112m (LPDAAC, 2013).

A resolução (nominal) de aproximadamente 90m dos dados SRTM disponíveis pode ser considerada razoável para análises em escalas pequenas ou médias, mas não pode ser aplicada em estudos de detalhe. Uma alternativa viável consiste em interpolar os dados originais com uma resolução maior (ou seja, para obter células de tamanho menor do que o original). Esta abordagem não irá aumentar o nível de detalhe do modelo resultante, porém resultará em uma superfície com coerência de suas propriedades angulares (p.ex., declividade, orientação de vertentes etc) entre as células vizinhas (Valeriano *et al.*, 2006), fator importante em análise morfométrica.

6.2. Download e importação dos dados

Agora que já vimos um pouco do básico sobre o GRASS, podemos prosseguir com a importação da imagem SRTM. O primeiro passo é determinar qual arquivo SRTM devemos baixar. Há várias opções de sites que disponibilizam dados SRTM, mas vamos utilizar o servidor do USGS. Ao abrir o endereço <http://e4ft101.cr.usgs.gov/SRTM/>, temos um conjunto de diretórios de acordo com o tipo dos dados (fig. 6.5). Informações mais detalhadas estão descritas no documento *The Shuttle Radar Topography Mission (SRTM) Collection User Guide*²⁰.

Short Name	Data Product	Rows/Columns	Spatial Resolution
SRTMGL1	Global 1 arc-second	3601/3601	1 arc-second
SRTMGL1N	Global 1 arc-second number	3601/3601	1 arc-second
SRTMGL3	Global 3 arc-second	1201/1201	3 arc-second
SRTMGL30	Global 30 arc-second	6000/4800	30 arc-second
SRTMGL3N	Global 3 arc-second number	1201/1201	3 arc-second
SRTMGL3S	Global 3 arc-second sub-sampled	1201/1201	3 arc-second
SRTMSWBD	Water Body Data Shapefiles & Raster files	3601/3601	1 arc-second
SRTMIMGR	Global SRTM Swath Image Data	3601/3601	1 arc-second
SRTMIMGM	Global SRTM Combined Image Data	3601/3601	1 arc-second

Figura 6.5. Características dos conjuntos de dados SRTM.

Vamos baixar dados com resolução de 01”, ou aproximadamente 30m. Entre no diretório SRTMGL1.003 e em seguida em 2000.02.11/. Pode demorar um pouco para carregar a listagem toda, afinal são mais de 14.000 arquivos! Cada arquivo representa uma área de 1°x1°e é nomeado de acordo com as coordenadas do canto esquerdo inferior. Assim um arquivo para a área entre 24°S, 25°S, 48°W e 47°W, o arquivo é S25W048. Se você já sabe qual arquivo quer, pode digitar diretamente o endereço completo:

```
http://e4ft101.cr.usgs.gov/SRTM/SRTMGL1.003/2000.02.11/S25W048.SRTMGL1.2.jpg      imagem preview
http://e4ft101.cr.usgs.gov/SRTM/SRTMGL1.003/2000.02.11/S25W048.SRTMGL1.hgt.zip      dados de elevação
http://e4ft101.cr.usgs.gov/SRTM/SRTMGL1.003/2000.02.11/S25W048.SRTMGL1.hgt.zip.xml  metadados
```

Para nossos exemplos, baixe os arquivos S25W048 e S25W049.

Após baixar os arquivos, continue com a importação. Como os dados estão em coordenadas geográficas e nossa location está em UTM, é preciso projetar os dados. Nas versões anteriores do GRASS esse processo era um pouco complicado: era preciso criar uma Location em UTM e outra no sistema dos dados originais (como Latitude-Longitude), importar os dados no sistema original, sair do GRASS, iniciar uma nova sessão na Location em UTM e projetar os dados de uma Location para a outra. Agora isso é feito automaticamente.

²⁰https://lpdaac.usgs.gov/sites/default/files/public/measures/docs/NASA_SRTM_V3.pdf

Vá em **File** → **Import raster data** → **Import of common formats with reprojection [r.import]**. Note que à medida em que você movimenta o ponteiro do mouse sobre os ítems dos menus, o nome do comando e uma breve descrição aparecem na parte inferior esquerda da janela do *Layer Manager*. Na figura 6.6 vemos as opções do módulo. Antes de importar, lembre-se de descompactar o arquivo que você baixou. Na aba **Input** selecione o arquivo S25W048.hgt como dado de entrada. Na aba **Output**, digite um nome para o arquivo de saída (S25W048 é uma boa ideia) e defina a opção **resolution** como **value** e a opção **resolution_value** como 30.



Figura 6.6. Opções do comando **r.import**.

Com as imagens importadas, vamos exibi-las no *Map Display*. Nas versões mais atuais do GRASS, vários comandos já se encarregam de incluir o *layer* resultante na pilha de *layers* do *Layer Manager*. Caso isso não ocorra, adicione um *layer* tipo *raster* clicando sobre o sexto ícone da barra de ferramentas superior do *Layer Manager* (fig. 5.13). Na aba **Required**, um menu tipo *drop-down* mostra uma lista com os *rasters* disponíveis (fig. 6.7). Selecione um dos arquivos que importamos e clique em **OK**. Faça o mesmo para o segundo arquivo.

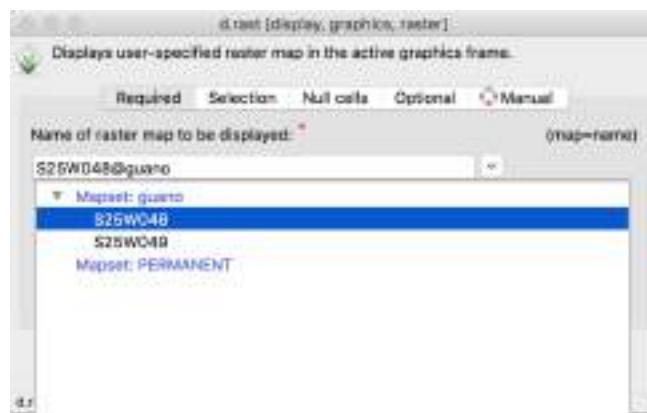


Figura 6.7. Insira um novo *layer* na pilha.

No Map Display, use as ferramentas de **Zoom in**, **Zoom out** e **Pan** (veja a figura 5.14) para enquadrar as duas imagens na tela. Se estiver usando os mesmos arquivos de exemplo, você deverá ver o mesmo que a figura 6.8. As cores representam a elevação do modelo, e rapidamente pode-se notar algumas áreas onde a variação de cores não é suave como seria de se esperar, mas é brusca (mais fácil de ver nas área da junção entre as duas imagens).

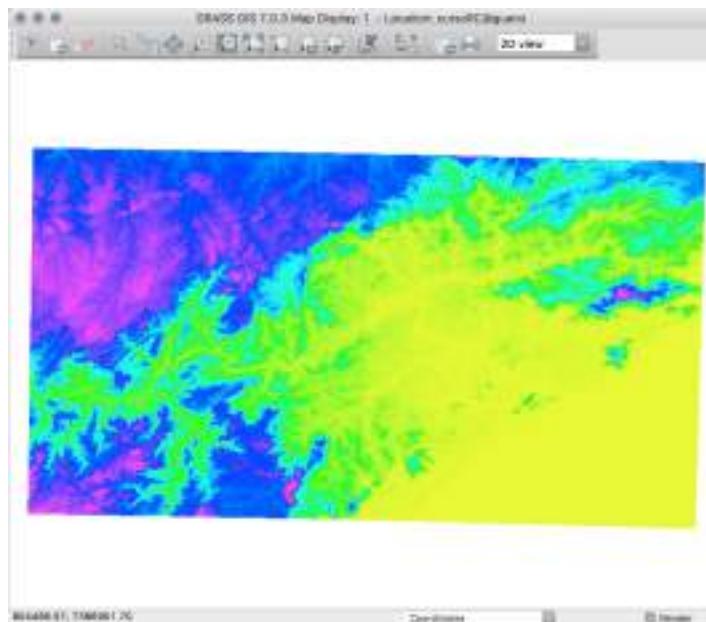


Figura 6.8. As imagens SRTM, exibidas no Map Display.

6.3. Raster patching

Nosso primeiro passo será juntar os dois arquivos SRTM em um raster único. Para isso iremos usar o comando `r.patch`, que você encontra em **Raster** → **Overlay rasters** → **Patch raster maps [r.patch]**.

Vamos aproveitar e ver algumas das opções que comumente temos nos comandos do GRASS. Quando a janela de diálogo de um comando é aberta, vemos algumas abas na parte superior (a quantidade de abas varia de acordo com o comando). Na figura 6.9 temos a primeira aba, **Required**, que como o nome indica, são os parâmetros obrigatórios para que o módulo possa ser executado. Na aba **Optional** (fig. 6.9) temos algumas opções, como deixar que o GRASS sobrescreva arquivos já existentes (o padrão é não sobrescrever). A aba **Command output** (fig. 6.10) irá mostrar as informações de saída do módulo e a aba **Manual** (fig. 6.10) traz explicações sobre todas as opções e funcionamento do comando.

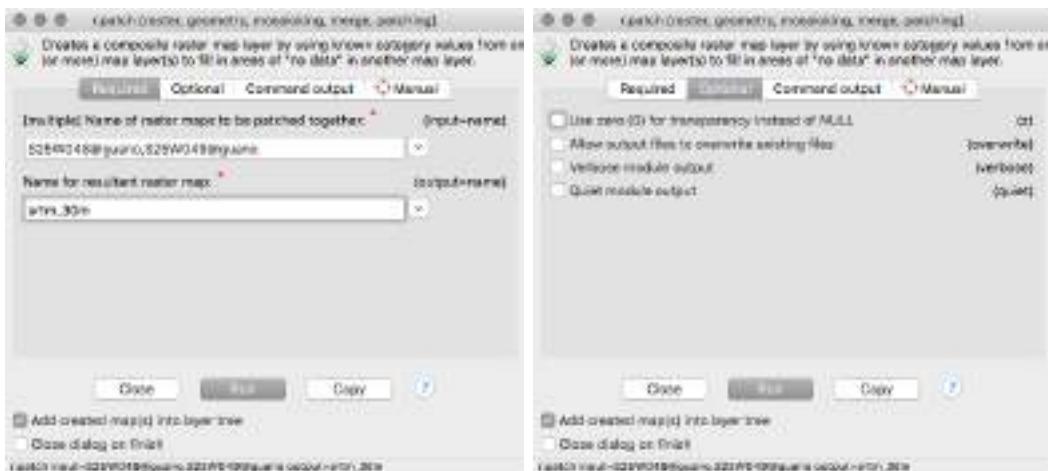


Figura 6.9. Abas **Required** e **Optional** do comando `r.patch`.



Figura 6.10. Abas **Command output** e **Manual** do comando `r.patch`.

Na aba **Required**, o primeiro campo que temos a preencher é com os mapas que queremos juntar. Selecione os dois rasters que importamos (um por vez). No segundo campo indique o nome do arquivo de saída, no nosso exemplo, usamos `srtm_30m`.

Aqui vale uma dica: computadores são burros, e ter que trabalhar em plataformas diferentes (Windows, Mac, Linux) só complica as coisas. Então tente minimizar as chances de problemas futuros e use nomes inteligentes em seus arquivos. Não use espaços entre as palavras, use um ‘_’ (*underscore/underline*) ou use o que programadores chama de ‘*CamelCase*’, onde cada palavra começa com letra maiúscula, sem espaço entre as palavras. Como os nomes de arquivos serão usados internamente pelos programas de SIG (ou seja, serão variáveis em termos de programação), os nomes não devem (não podem) começar por números, mas podem ter números no meio ou no final do nome. Finalmente, tente usar nomes que indiquem que arquivo é esse e o que foi feito com ele, afinal nossa memória não é tão confiável a longo prazo. Um raster que representa um modelo de terreno pode muito bem ser chamado de **DEM** (*Digital Elevation Model*) ou **MDT** (*Modelo Digital de Terreno*) como em `DEM_petar` ou `petarDEM`. Se esse raster passar por uma filtragem, por exemplo, o arquivo de saída pode ser `petarDEM_filter_3x3` (aqui 3x3 representa o tamanho do filtro aplicado). Bem melhor do que nomear os arquivos de `mapa1`, `mapa2`, `mapa_teste`, `mapa_agora_vai...`

Após rodar o comando, o novo raster `srtm_30m` já deve ser carregado e exibido no *Map display* (fig. 6.11). Veja que agora a transição de cores é suave, pois antes uma tabela de cores era aplicada a cada layer, e agora temos apenas um layer. Desligue a visualização dos dois rasters que importamos inicialmente e veja que o mapa gerado pelo `r.patch` ficou “cortado” na porção Leste. Para entender porque isso aconteceu, precisamos falar do conceito de **region**.

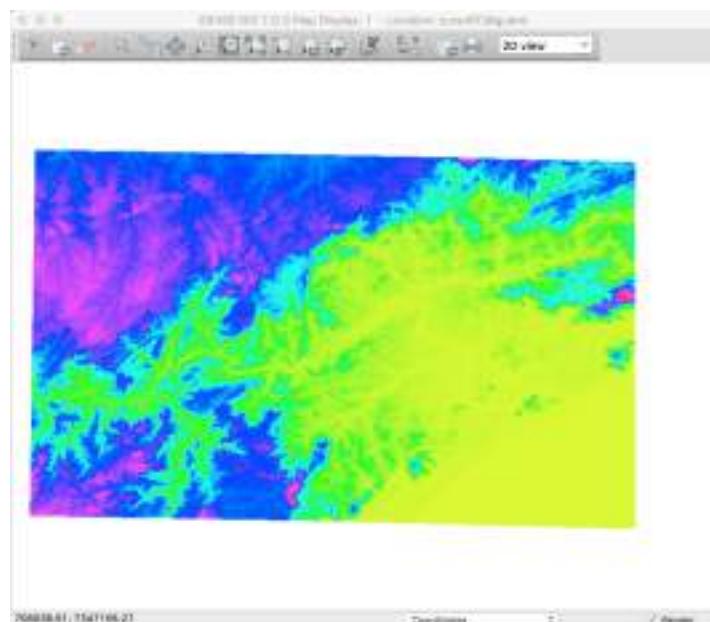


Figura 6.11. Mapa raster resultante do comando `r.patch`.

Antes disso vá ao *Layer Manager* e clique sobre o ícone que está logo à direita do nome do layer do arquivo que criamos (clicar com o botão direito do mouse sobre o nome do arquivo também funciona). Veja que aqui temos várias opção à disposição, como remover o arquivo da pilha (sem deletar o arquivo do computador), renomear o nome de exibição (o arquivo não é renomeado), mudar a opacidade (deixar o mapa transparente), mudar as propriedades gráficas do que é exibido (cores, espessura de linhas etc) e várias outras.

No momento, clique na última opção, *Metadata*. Os *metadados* são dados sobre os dados, e indicam propriedades importantes, como limites geográficos, hora de criação do arquivo e histórico de ações (como indicar que o arquivo foi criado a partir da junção de dois outros arquivos). Ao selecionar essa opção, o *Layer Manager* deve ativar a aba **Command console**, com a saída do comando `r.info`.

Esta é a informação que foi exibida na janela do *Layer Manager*:

```
+-----+
| Map:      srtm_30m@guano           Date: Wed May 18 21:31:50 2016   |
| Mapset:   guano                  Login of Creator: guano          |
| Location: cursoRC               |                                 |
| DataBase: /Volumes/MacintoshHD2/grassdata |                         |
| Title:    ( srtm_30m_full ) |                         |
| Timestamp: none                |                         |
+-----+
|
|
| Type of Map: raster           Number of Categories: 0   |
| Data Type:   CELL             |                         |
| Rows:        4284             |                         |
| Columns:    7271              |                         |
| Total Cells: 31148964        |                         |
| Projection: UTM (zone -22) |                         |
|      N: 7351140   S: 7222620   Res: 30 |                         |
|      E: 915270    W: 697140    Res: 30 |                         |
| Range of data: min = -50  max = 1463 |                         |
|
|
| Data Description:           |
|   generated by r.patch      |
|
| Comments:                   |
|   r.patch input="S25W048@guano,S25W049@guano" output="srtm_30m_full" |
+-----+
```

6.4. Trabalhando com `regions`

O conceito de `region` é exclusivo do GRASS, e muito poderoso. A `region` define quais os limites da área geográfica sendo utilizada para análises e qual a resolução espacial que será adotada.

Ao alterar a resolução da `region`, o GRASS faz uma reamostragem dos dados pelo método do vizinho mais próximo, ou seja, o valor a ser usado para um pixel será o valor do pixel original mais próximo de seu centro.

Alterar os limites da `region` permite, por exemplo, executar análises em sub-áreas de imagens que são muito grandes e cujo processamento seria muito demorado. Assim podemos testar nossas ideias rapidamente, antes de aplicá-las para uma área maior.

Vamos ver como está definida nossa `region`. Na janela de terminal (ou na aba de *Command Console* do *Layer Manager*), digite `g.region -p` e pressione `Enter`:

```
GRASS 7.0.3 (cursorRC):~ > g.region -p
projection: 1 (UTM)
zone:      -22
datum:     wgs84
ellipsoid: wgs84
north:     7405000
south:     7160000
west:      640000
east:      885000
nsres:    29.99877556
ewres:    29.99877556
rows:      8167
cols:      8167
cells:    66699889
```

Lembre-se que existe um espaço em branco entre o comando (`g.region`) e a *flag* (opção) `-p`, mas não entre o traço (`-`) e a letra “`p`”. Várias flags podem ser agrupadas diretamente, como em `-pae` ao invés de `-p -a -e`.

Aqui o comando `g.region` imprime (com a opção `-p`) as configurações da `region` ativa: os limites geográficos, a resolução espacial e a quantidade de linhas e colunas.

Essa é a `region default`, que definimos quando criamos a `Location`. Veja que apesar de termos definido a resolução espacial como 30m, os valores de `nsres` e `ewres` não são exatamente 30m, porque o GRASS ajustou o tamanho das células (ou *pixels*) para encaixar nos limites geográficos. Se você quiser que a resolução seja fixa, mas que os limites possam ser levemente alterados, use a *flag* `-a` do comando `g.region` juntamente com o parâmetro `res=30`:

```
GRASS 7.0.3 (cursoRC):~ > g.region -pa res=30
projection: 1 (UTM)
zone: -22
datum: wgs84
ellipsoid: wgs84
north: 7405020
south: 7159980
west: 639990
east: 885000
nsres: 30
ewres: 30
rows: 8168
cols: 8167
cells: 66708056
```

Agora vamos ver esses limites da `region` no *Map Display*. Na porção inferior da janela do *Map Display*, há um menu *drop-down* com algumas opções (fig. 6.12). O padrão é estar na opção **Coordinates**, o que exibe as coordenadas da posição do cursor na parte esquerda inferior do *Map Display*. habilite a opção **Show comp. extent**, e veja que deve aparecer um retângulo em volta do nosso mapa (fig. 6.13). Esse retângulo indica os limites da `region`, e sua cor é azul quando o display está encaixado na region e vermelho nas outras situações.

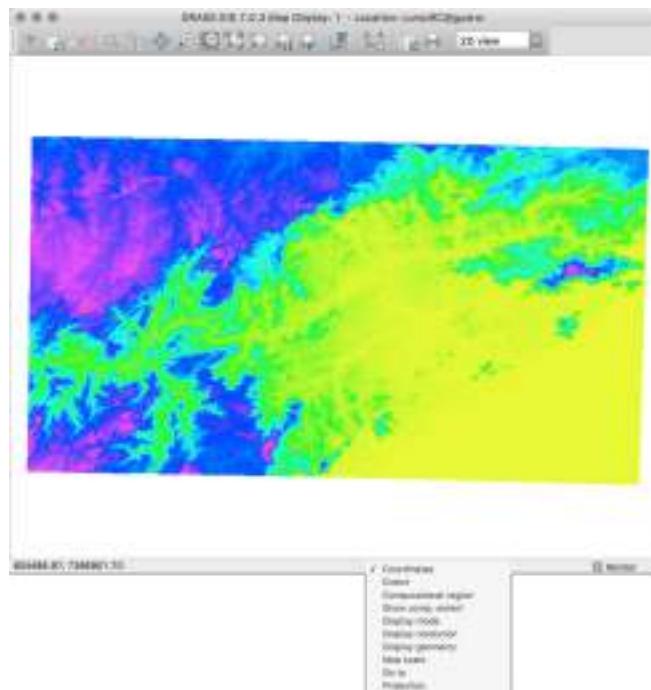


Figura 6.12. Opções do *Map Display*.

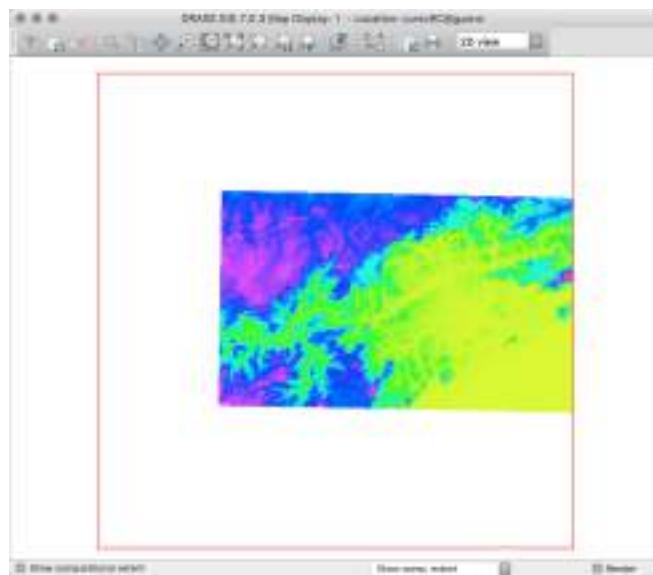


Figura 6.13. *Map Display* com exibição dos limites da **region**.

Isso explica porque o resultado do comando `r.patch` ficou “cortado”: os comandos do GRASS que trabalham com rasters (e imagens), sempre obedecem os limites da `region`. Então quando o `r.patch` calculou o mapa novo a partir dos dois anteriores, ele fez isso apenas dentro dos limites da `region`.

Vamos rodar o `r.patch` novamente, mas antes vamos alterar os limites da region. Habilite a visualização dos dados SRTM no *Layer Manager* e use as ferramentas de zoom para que os dois raster estejam visualizados, tal como na figura 6.8. Em seguida clique no botão **Various zoom options** na barra de ferramentas do *Map Display* (o último dos botões com a lupa de zoom) e selecione **Set computational region extent from display** (fig. 6.14).

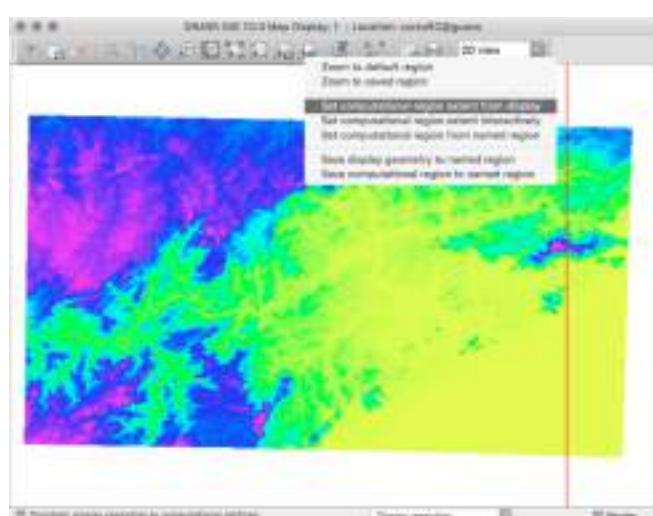


Figura 6.14. Definindo os limites da `region` a partir da área exibida no *Map Display*.

Agora o retângulo da `region` engloba todos nossos dados. Veja como ficaram as definições dos limites com `g.region`:

```
GRASS 7.0.3 (cursoRC):~ > g.region -p
projection: 1 (UTM)
zone:      -22
datum:     wgs84
ellipsoid: wgs84
north:    7370430
south:    7204710
west:     697740
east:     910890
nsres:    30
ewres:    30
rows:     5524
cols:     7105
cells:    39248020
```

Rode o comando `r.patch` novamente, e use o mesmo nome `srtm_30m` para o arquivo de saída. Será preciso escolher a opção **Allow output files to overwrite existing files**, ou o GRASS irá retornar um erro dizendo que o arquivo de saída já existe, e não vai rodar o comando. O resultado final deverá estar como na figura 6.15, sem aquele “corte” que observamos anteriormente.

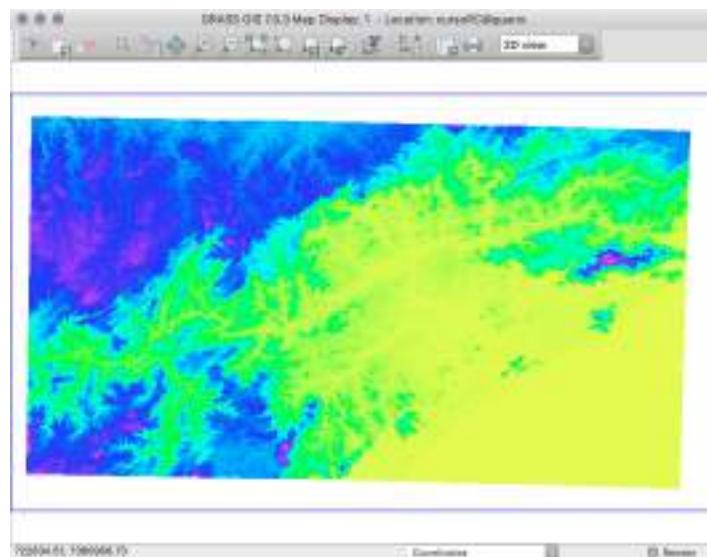


Figura 6.15. Novo mapa raster resultante do comando `r.patch`.

Agora vamos ver o efeito de alterar a resolução espacial da `region`. Antes vamos definir uma `region` para coincidir com os limites e resolução do raster `srtm_30m`:

```
GRASS 7.0.3 (cursoRC):~ > g.region raster=srtm_30m -p
projection: 1 (UTM)
zone:      -22
datum:     wgs84
ellipsoid: wgs84
north:    7351140
south:    7222620
west:     697140
east:     915270
nsres:    30
ewres:    30
rows:     4284
cols:     7271
cells:    31148964
```

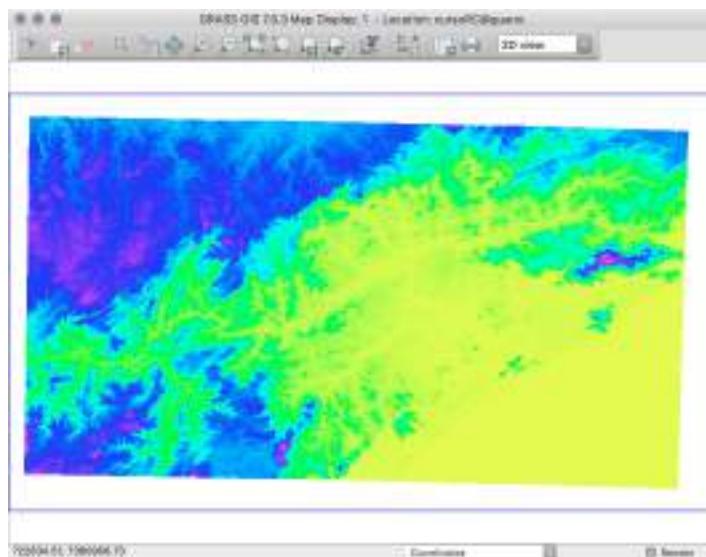


Figura 6.16. `region` ajustada ao limite do *raster* criado com `r.patch`.

O resultado no *Map Display* não foi exatamente o que você esperava? Se os limites da `region` ficaram muito longe das áreas do mapa e você esperava que eles ficassem bem ajustados aos dados, isso é porque quando criamos o *raster* com `r.patch`, os limites da `region` eram esses, então o GRASS entendeu que toda essa área deveria fazer parte do *raster*. A porção em volta dos dados, em que não vemos nada, está definida como de valor nulo (NULL, na linguagem do GRASS). Se quisermos ajustar a `region` aos dados não-nulos, usamos a opção `zoom`:

```
GRASS 7.0.3 (cursoRC):~ > g.region zoom=srtm_30m -p
projection: 1 (UTM)
zone:      -22
datum:     wgs84
ellipsoid: wgs84
north:     7344330
south:     7229070
west:      701820
east:      907080
nsres:    30
ewres:    30
rows:     3842
cols:     6842
cells:    26286964
```

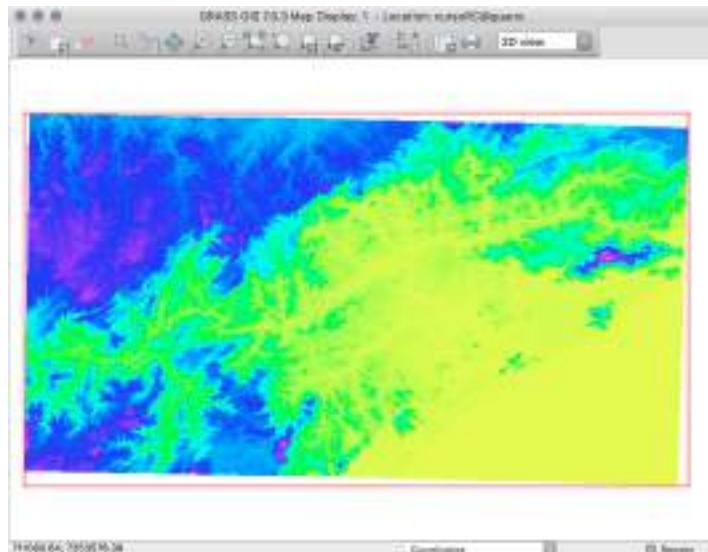


Figura 6.17. `region` ajustada ao limite do dados não-nulos do *raster* criado com `r.patch`.

Mas ainda temos uma área muito grande para trabalhar. Vamos escolher uma sub-área interativamente. No **Map Display**, usa a ferramenta de **Zoom** e selecione uma área. Depois de escolher a sub-área, vá em **Zoom options** → **Set computational region extent from display**. Depois, no terminal, ajuste a resolução para 30m, forçando o tamanho dos pixels usando `g.region -pa res=30`.

Assim você pode selecionar uma área interativamente até que fique do seu agrado. E pode ainda salvar essa `region` para acesso rápido, com `g.region -pa res=30 save=petar`. Para ativar a `region` já salva, use `g.region region=petar`.

Podemos ver como os parâmetros da `region` afetam nossos dados se configurarmos o *Map Display* para exibir os mapa de acordo com as configurações da `region` ativa (por padrão, os mapas raster são exibidos com suas propriedades originais).

Para isso, localize na barra inferior do *Map Display* um menu de opções. Selecione **Display resolution** e veja que na parte esquerda da barra agora há uma *check-box* que indica **Constrain display resolution to computational settings**. Se essa caixa estiver marcada, a resolução dos mapas no display irá seguir as definições da `region` (os dados não são alterados). Faça o teste, marque a caixa, altere a resolução da `region` e force o re-desenho dos dados usando o segundo ícone da barra de ferramentas do Map Display (**Render map**). As figuras a seguir mostram esse comportamento.

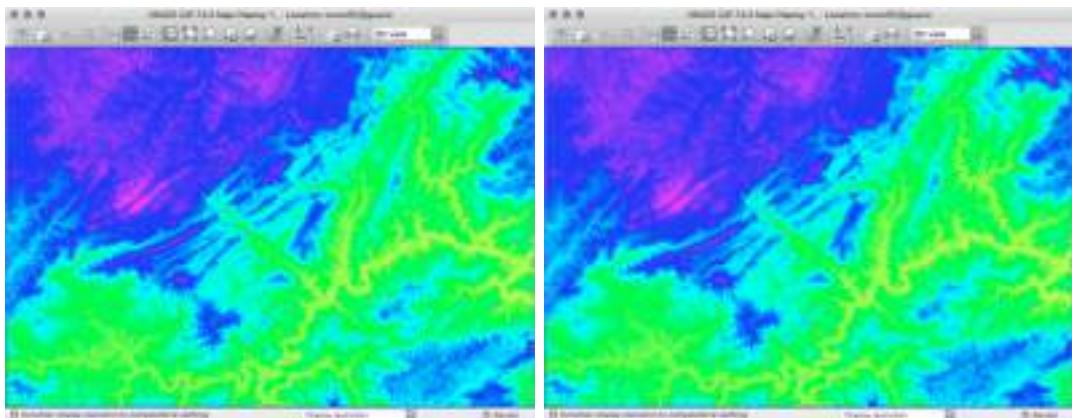


Figura 6.18. Display com resolução de 30m (esquerda) e 300m (direita).

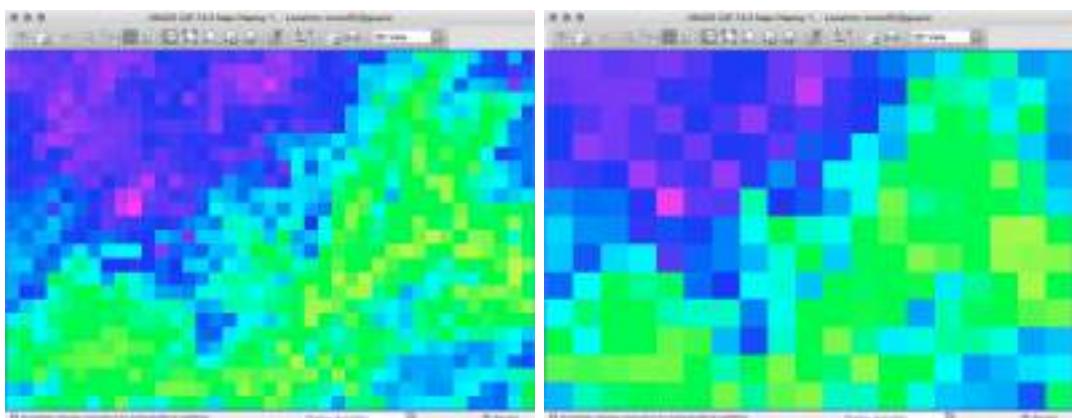


Figura 6.19. Display com resolução de 1500m (esquerda) e 3000m (direita).

7. Dados vetoriais

Neste capítulo veremos um pouco mais sobre dados em formato vetorial. Vamos aproveitar a oportunidade para começar a trabalhar com o QGIS.

7.1. Arquivos ESRI shape

Dentre os diversos formatos de mapas vetoriais existentes, provavelmente o mais utilizado atualmente seja o *shapefile*, criado pela empresa ESRI²¹, desenvolvedora do pacote SIG proprietário ArcGIS[©]. Com uma implementação simples e especificação técnica aberta²², é muito difícil encontrar, hoje em dia, programas SIG que não suportem o formato SHP.

Um *shapefile* consiste de um arquivo principal, um arquivo de índice e uma tabela em formato dBASE. O arquivo principal contém os registros de todos os elementos gráficos (*shapes* – pontos, linhas, polígonos) com uma lista dos vértices de cada elemento. No arquivo de índice, cada registro determina a posição do registro correspondente no arquivo principal. A tabela dBASE contém os atributos das entidades geométricas. Note que devido à limitações do formato dBASE IV, os nomes das colunas na tabela de atributos não podem exceder *onze* caracteres.

Adicionalmente, pode-se ter um arquivo que especifique a projeção cartográfica e ainda arquivos auxiliares. Todos os arquivos devem ter o mesmo nome, sendo diferenciados pela extensão. Por exemplo:

estradas.shp – arquivo principal, armazena os elementos gráficos

estradas.shx – arquivo de índice dos elementos gráficos

estradas.dbf – tabela de atributos em formato dBASE IV

estradas.prj – arquivo de projeção cartográfica

estradas.sbn – arquivo auxiliar - índice espacial dos elementos gráficos

estradas.sbx – arquivo auxiliar - índice espacial dos elementos gráficos

7.2. Dados vetoriais no QGIS

Um dado muito interessante é o mapa-índice que mostra a situação atual do mapeamento sistemático topográfico do Brasil. São diversos layers vetoriais, que trazem quais mapas topográficos estão disponíveis de acordo com a escala. Baixe o arquivo em ftp://geoftp.ibge.gov.br/mapeamento_sistemático/mapa_indice_digital_4ed/produto_mapa_indice_digital/shape.zip. A nota explicativa está em ftp://geoftp.ibge.gov.br/mapeamento_sistemático/mapa_indice_digital_4ed/leia_me_4ed.pdf.

²¹<http://www.esri.com>

²²<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

Após baixar os dados, descompacte o arquivo `shape.zip` e inicie o QGIS. A interface do QGIS possui uma área à esquerda para a pilha de *layers* carregados e uma área à direita para a exibição dos mapas.

Tanto no QGIS como na maioria dos SIG (ArcGIS etc) os dados (shapefiles, geotiffs etc) não são abertos usando a tão comum combinação **File → Open**. Esse caminho é usado para abrir arquivos de **projeto**, que são usados quando se deseja salvar todo o esquema de trabalho, incluindo os diversos *layers* carregados, suas propriedades gráficas (cores, simbologia) e demais configurações. Os dados vetoriais ou raster são **adicionados** no projeto atual. Isso é feito clicando no ícone com um símbolo de “+”(*Add vector layer* - fig. 7.1). O QGIS tem botões diferentes para cada tipo de dado (vetor, raster etc), enquanto que o ArcGIS, por exemplo, tem apenas um botão para adicionar dados, independente do tipo.

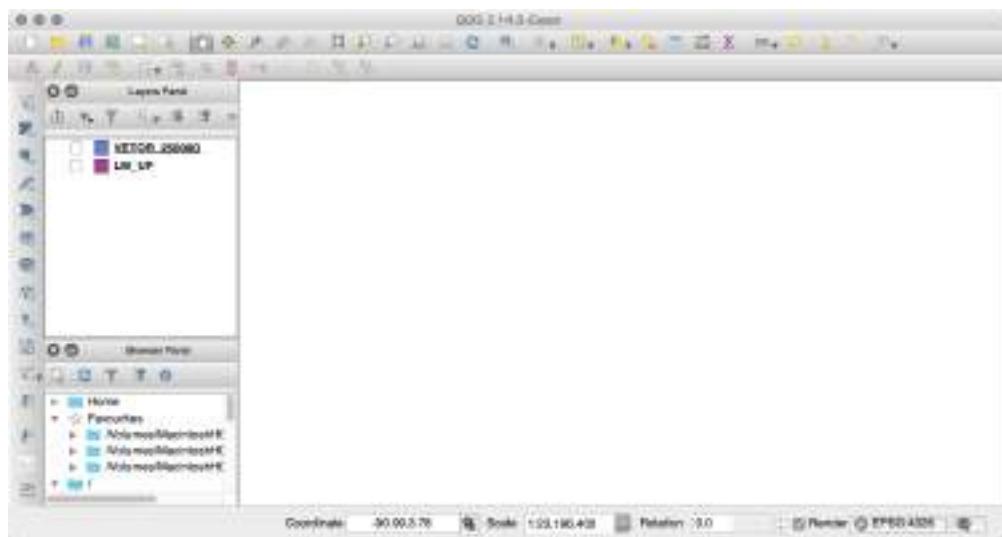


Figura 7.1. Interface gráfica do QGIS.

No arquivo `shape.zip` que baixamos, existem alguns dados de referência além das cartas topográficas disponíveis por escala. São dados em escala 1:1.000.000:

- Malha Municipal (`LM_MUNICIPIO_2007`)
- Hidrografia (`HD_MID_CURSO_DAGUA` e `HD_MID_MASSA_DAGUA`)
- Rodovias (`ST_MID_RODOVIA`)
- Capitais (`LC_MID_CAPITAL`)
- Unidades federativas (`LM_UF`)

Adicione, por exemplo, o layer **LM_UF** e você terá um mapa do Brasil dividido por estado (cada estado é um polígono). Adicione depois o layer **VETOR_250000**, que mostra quais mapas topográficos em escala 1:250.000 estão disponíveis em formato vetorial (fig. 7.2). As cores que você verá certamente serão diferentes das exibidas nas figuras, pois elas são atribuídas aleatoriamente.

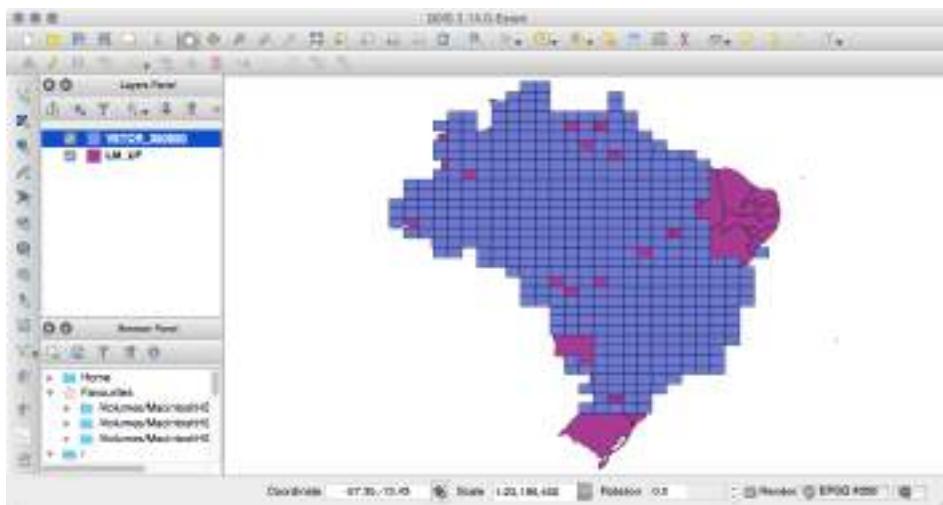


Figura 7.2. QGIS com layers LM_UF e VETOR_250000 carregados.

Aproveite para explorar a interface do QGIS. O ícone com um “i” é o da ferramenta para identificar feições e mostrar os atributos dessas feições que estão na tabela de atributos. O layer que será analisado é o que estiver selecionado na pilha de layers. A seleção é feita ao clicar sobre o *nome* do layer, e uma faixa azul sobre o nome mostra qual layer está selecionado. No nosso exemplo, selecionamos o layer VETOR_250000, e ao clicarmos sobre um dos polígonos vemos uma janela como a da figura 7.3. Todas essas informações estão armazenadas na tabela de atributos (o arquivo .dbf) do shapefile.

Figura 7.3. Informações sobre um dos polígonos do layer VETOR_250000 (parcial).

Mas fica um pouco difícil visualizar os dois layers um sobre o outro. vamos então alterar as propriedades de exibição do layer VETOR_250000. Dê um duplo-clique sobre o nome do layer (ou clique com o botão direito e selecione a opção **Properties**). A janela de diálogo das propriedades é mostrada na figura 7.4. Aqui podemos, por exemplo, alterar a cor do preenchimento dos polígonos, a cor e espessura das linhas entre várias outras opções. Altere a transparência do preenchimento dos polígonos e clique em **Apply**. Assim você pode testar várias configurações até encontrar a que mais lhe agrada. Quando estiver satisfeito, clique em **OK**.

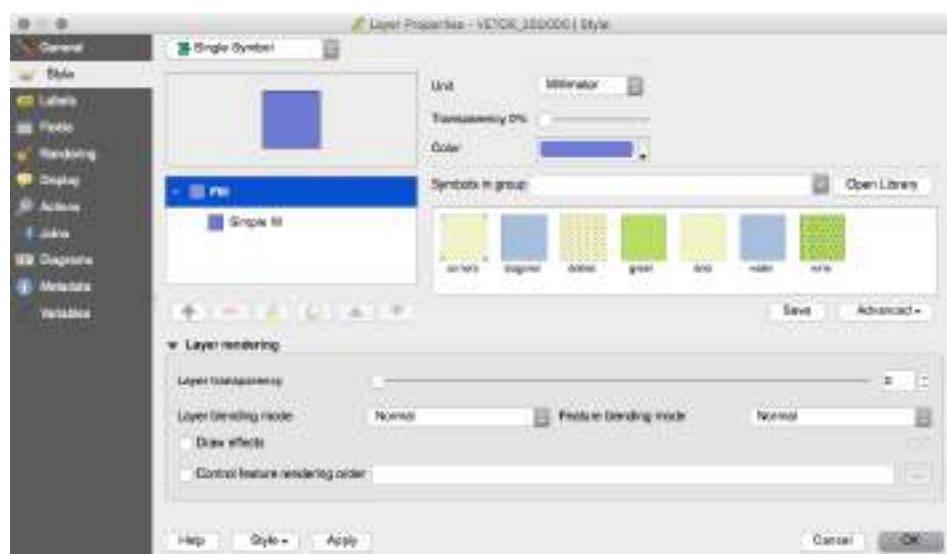


Figura 7.4. Janela de diálogo das propriedades do layer.

Na figura 7.5 vemos que alguns dos polígonos estão mais escuros que outros. Qual será a causa? Como os preenchimentos estão com transparência, uma cor mais escura pode indicar que há dois ou mais polígonos sobrepostos, o que nesse caso indicaria um erro nos dados, pois há duplicação de informação. Para confirmar a suspeita, use a ferramenta de informação e clique sobre um desses polígonos mais escuros (pode ser necessário usar a ferramenta **Zoom** antes). Na janela de resultados (7.6) vemos que realmente temos dois elementos no mesmo lugar.

Muito bem. Agora que já estamos acostumados com o QGIS, vamos baixar alguns dados. Como estamos trabalhando com a área do Vale do Ribeira, vamos usar dados do Sistema de Informações Geográficas do Ribeira de Iguape e Litoral Sul (<http://www.sigrb.com.br>). É necessário fazer um cadastro gratuito para baixar os dados. No site do SIG-Ribeira, vá em **Download → Arquivos** e escolha **Dados SIG-RB 3.1 - Vetores e Matrizes**. O arquivo tem cerca de 360MB em formato ZIP e quase 1,5GB descompactado. Dentro do arquivo, localize os diretórios **Drenagem**, **Isolinhas** e **Pontos_Cotados**.

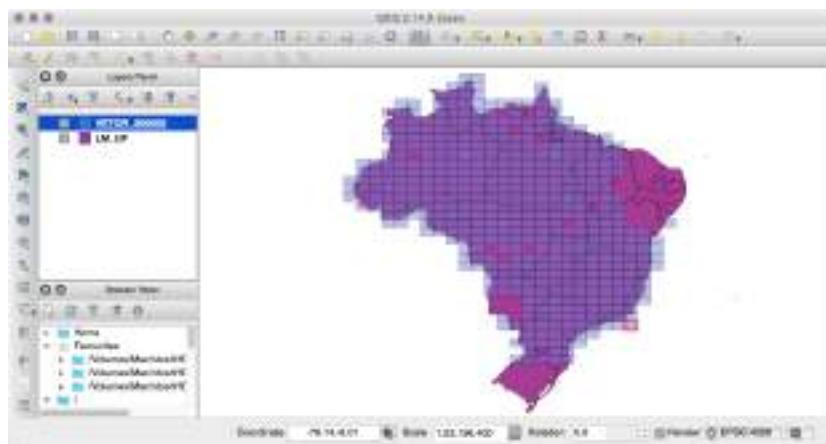


Figura 7.5. Layer VETOR_250000 com transparência no preenchimento.

Feature	Value
VETOR_250000	
ID	150
ID	456

Figura 7.6. Janela de informações com dois resultados.

No QGIS, inicie um novo projeto (o primeiro ícone da barra de ferramentas) e insira os shapes selecionados do SIG-Ribeira. Esses dados representam a hidrografia e topografia de todo o vale do Rio Ribeira de Iguape (7.7), digitalizados a partir de mapas topográficos 1:50.000.

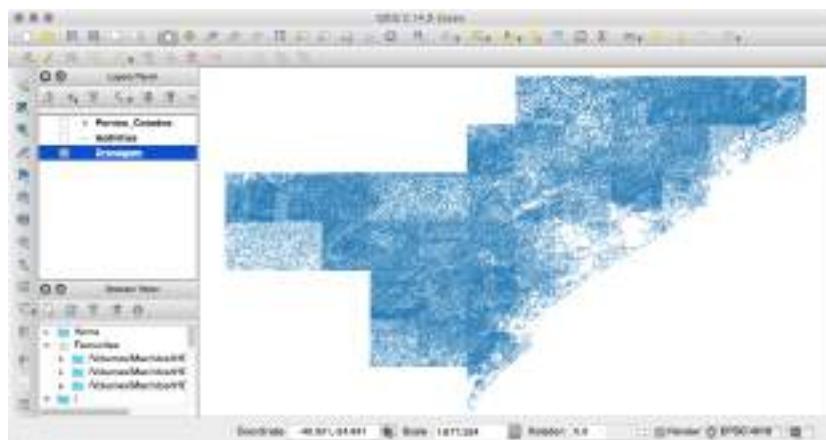


Figura 7.7. Dados de hidrografia do projeto SIG-Ribeira.

Abra a janela de propriedades de um dos arquivos e na aba **General**, veja as informações em **Coordinate reference system**. O arquivo está em coordenadas geográficas e com *datum*

SAD69. Isso significa que será necessário projetar esses dados para importá-los no GRASS. Aproveite e anote o código EPSG dessa projeção: 4618.

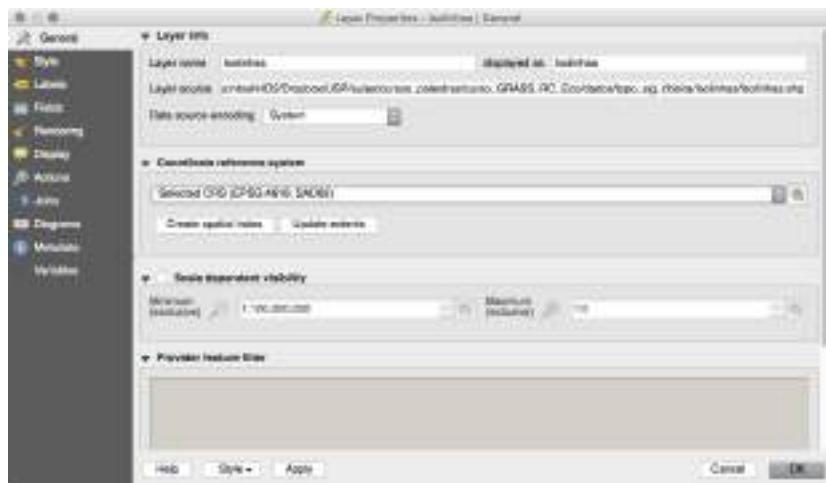


Figura 7.8. Sistema de coordenadas dos dados do SIG-Ribeira.

Outra maneira de determinar o sistema de coordenadas é abrir o arquivo de extensão `.prj` que acompanha o arquivo shapefile:

```
GEOGCS["GCS_South_American_1969",
DATUM["D_South_American_1969",
SPHEROID["GRS_1967_Truncated",
6378160.0,298.25]],
PRIMEM["Greenwich",0.0],
UNIT["Degree",0.0174532925199433]]
```

7.3. Arquivos ASCII

A importação de pontos vetoriais a partir de arquivos texto é uma tarefa bastante comum em SIG. São arquivos de texto *puro* (também chamado de **ASCII**²³), sem formatação (negrito, itálico, etc). Podem ter extensão `.txt` (*text*), `.dat` (*data*), `.csv` (*comma separated values*), `.asc`, entre outras. Os dados são dispostos um por linha, com os atributos (latitude, longitude, elevação, etc) separados por espaço, vírgula, tabulação, ou outros caracteres especiais, como “&” e “|”.

Ao invés de simplesmente importar um arquivo XYZ qualquer, vamos aproveitar esta oportunidade para ver mais alguns comandos do GRASS. Vamos criar um arquivo de pontos vetoriais aleatoriamente distribuídos em uma área, atribuir um valor de elevação para cada ponto (a partir do modelo SRTM), exportar esses pontos em formato ASCII e finalmente importar o arquivo XYZ.

²³<http://pt.wikipedia.org/wiki/ASCII>

7.3.1 Geração de pontos aleatórios

Primeiramente, vamos ajustar os limites da `region` ativa para aquela que salvamos anteriormente. No terminal digite `g.region -p region=petar`.

Se quiser conferir se a operação funcionou, use a opção **Zoom to...** → **Zoom to computational region** e confira se a área exibida é a mesma que foi escolhida antes.

Agora vamos criar os pontos aleatórios, com o comando `v.random`. Vá em **Vector** → **Generate points** → **Generate random points [v.random]**. Crie um novo layer chamado `pontos_rand`, com 1000 pontos.

Certamente seu resultado será diferente da figura 7.9, uma vez que a posição dos pontos é definida aleatoriamente. Se você não gostar da distribuição dos pontos por algum motivo, como áreas de grande concentração de pontos e outras meio “vazias”, selecione a opção **Allow output files to overwrite existing files**, na aba **Optional**, e rode o comando novamente, até atingir um resultado satisfatório.

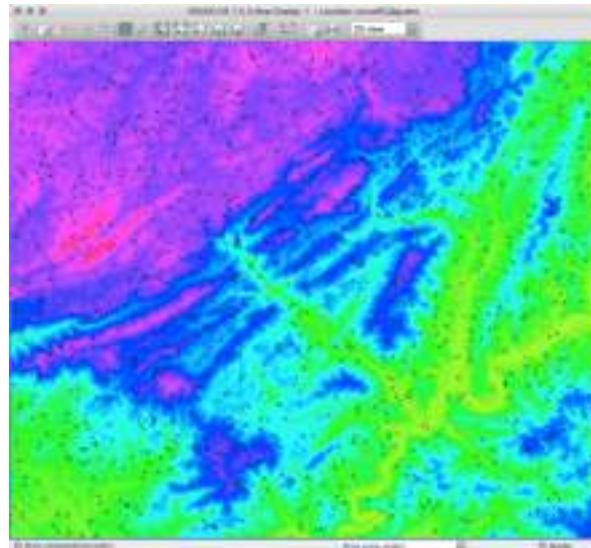


Figura 7.9. Pontos aleatórios gerados com `v.random`.

7.3.2 Amostragem da altitude

Agora vamos amostrar a altitude de cada ponto, a partir do modelo SRTM. Com o mapa `pontos_rand` selecionado no *Layer Manager*, procure a opção `Show attribute data`, e clique no ícone correspondente. Você deverá ver a mensagem de erro (Fig. 7.10):

O que significa que o mapa foi criado, mas a tabela de atributos não. Quando mapas vetoriais são criados dessa maneira, é preciso criar a tabela de atributos e povoá-la com uma linha por categoria. Isto pode ser feito com o comando `v.db.addtable`. Vá em **Databases** → **Vector database connections** → **New table [v.db.addtable]**.



Figura 7.10. Mensagem de erro do gerenciador de tabelas de atributos do GRASS.

Na aba **Required**, coloque `pontos_rand` como nome do mapa vetorial. Na aba **Definition** o primeiro campo pode ficar vazio (o nome do mapa será usado para o nome da tabela) e no último campo (**columns**) já podemos criar a coluna que vai armazenar os dados de elevação. Este campo é preenchido com o nome da coluna seguido do tipo de dados (*integer* para números inteiros, *float* para valores decimais e *string* para textos). Coloque `elev integer` e rode o comando.

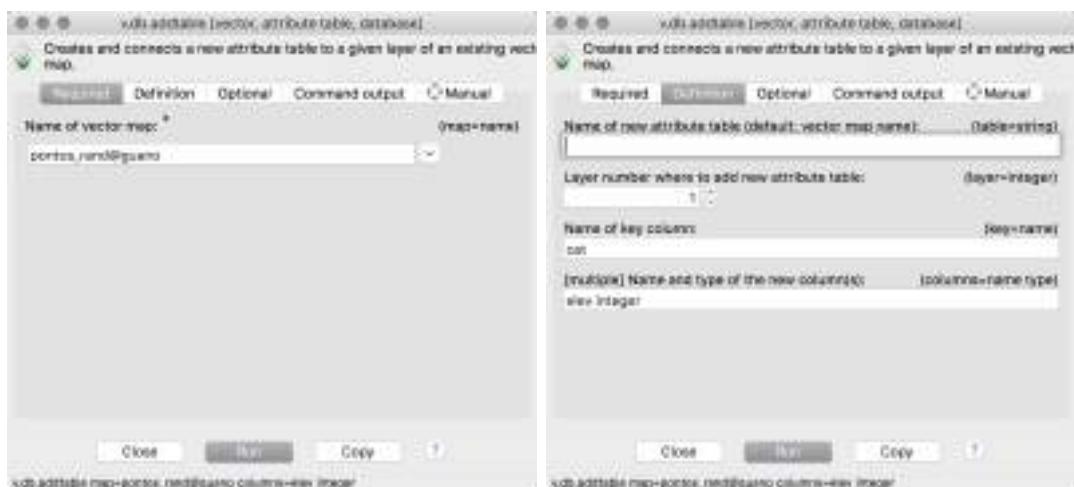


Figura 7.11. Opções do comando `v.db.addtable`.

Se você abrir o gerenciador de tabelas do GRASS verá uma interface tipo planilha, com uma linha para cada objeto do layer e suas colunas de atributos. Por enquanto nossa tabela tem a coluna `cat`, com valores sequenciais para os pontos, e a coluna `elev` está vazia.

Agora sim podemos extrair a altitude de cada ponto e vamos fazer isso de duas maneiras, incluindo valores na tabela e depois criando atributos de elevação 3D que não são armazenados na tabela.

Abra o comando `v.what.rast` em **Vector** → **Upadte attributes** → **Sample raster maps at point locations [v.what.rast]**. No diálogo do comando `v.what.rast`, escolha `pontos_rand` como mapa vetorial de entrada, `srtm_30m` como raster a ser questionado e na aba **Optional**, `elev` como a coluna do banco de dados que será atualizada com os valores do raster em cada ponto.

Abra novamente o gerenciador de tabelas e veja que agora as duas colunas de dados estão populadas, uma com a categoria de cada ponto e outra com o valor de elevação extraído do SRTM (fig. 7.13).



Figura 7.12. Coluna `elev` atualizada com os valores do SRTM.

Agora vamos usar o comando `v.drape`. Vá em **Vector → Develop vector map → Create 3D vector over raster [v.drape]**. Na aba **Required**, especifique como mapa de entrada `pontos_rand`, `pontos_rand_Z` como mapa resultante e escolha `srtm_30m` como raster a ser amostrado. Na aba **Selection**, deixa apenas `point` selecionado em **Input feature type**. O mapa gerado por este comando é um vetor 3D, ou seja, os valores amostrados do raster são armazenados como **elevação**, e não como atributos na tabela.

Após rodar este comando, se você usar a ferramenta **Query** do Map Display (**Query raster/vector map(s) (display mode)**), ao clicar sobre os pontos do mapa, você verá na aba **Command console** do *Layer Manager*, as propriedades de cada um, como coordenadas, categoria e elevação (fig. 7.13).



Figura 7.13. Atributos do ponto selecionado.

7.3.3 Exportação como ASCII

Com o mapa de pontos criado e convertido para 3D, podemos exportá-lo como ASCII, com o comando `v.out.ascii`. Vá em **File** → **Export vector Map** → **ASCII points or GRASS ASCII vector export [v.out.ascii]**. Na aba **Required** defina `pontos_rand_Z` como input e na aba **Output**, `pontos_rand_Z_ascii.txt` como resultado (output). Na aba **Selection**, deixe o tipo de formato de saída como `point`.

Se você não definir um diretório específico para o arquivo de saída, ele será gravado no seu diretório atual (descubra qual é usando `pwd` no terminal).

Apesar de em sistemas Linux/Mac OSX não ser exatamente obrigatório definir uma extensão para o arquivo (pois o sistema “sabe” que se trata de um arquivo texto ASCII e vai usar um editor de textos simples para abri-lo), no Windows é necessário indicar uma extensão, como `.txt`. Como boa prática, use uma extensão que descreva seus dados, como `.txt` ou `.asc`.

Abra o arquivo `pontos_rand_Z` com um editor de textos (Notepad, por exemplo). Você verá as colunas separadas pelo sinal “|”, como `X | Y | Z | cat`:

```
740528.99987136|7275840.86897838|80|1  
733448.47677301|7298806.06238889|967|2  
742406.78835657|7273536.74565529|398|3  
745103.32662358|7292579.89929052|474|4  
716935.08292914|7275286.76277608|197|5  
...
```

7.3.4 Importação do arquivo ASCII

Vamos agora importar o arquivo ASCII (nossa objetivo inicial!), com o comando `v.in.ascii`. Vá em **File** → **Import vector data** → **ASCII points or GRASS ASCII vector import [v.in.ascii]**.

Na aba **Required** defina `pontos_rand_Z.txt` como entrada e `pontos_ascii_import` como resultado. Na aba **Points**, use a coluna 3 como valor Z, para que o mapa seja criado como 3D. Note as opções *Do not create table in points mode* e *Do not build topology in points mode*, muito usadas quando o arquivo tem muitos pontos (da ordem de dezenas de milhares a milhões), o que torna inviável trabalhar com tabelas associadas à arquivos.

Por fim, exiba os dois mapas de pontos sobrepostos, no *Layer Manager*. Faça com que `pontos_rand_Z` fique por baixo. Altere as cores de linha e do preenchimento do símbolo para vermelho e mude tamanho do símbolo para 10. Altere o símbolo de `pontos_ascii_import` para `basic/diamond`, com tamanho 5 (fig. 7.14).

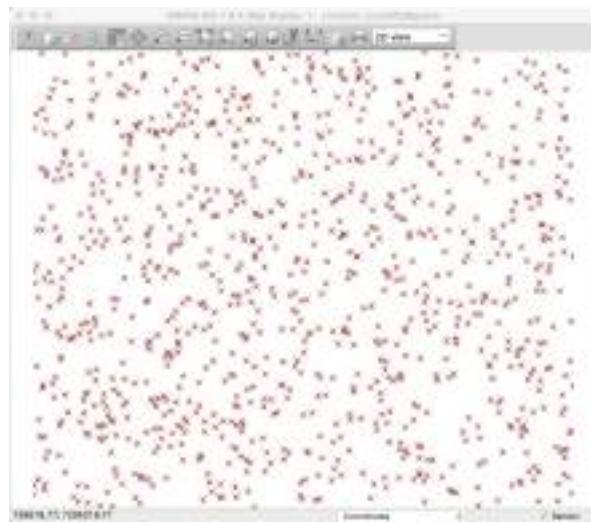


Figura 7.14. Mapa de pontos importado com `v.in.ascii` sobreposto ao mapa gerado com `v.random`.

7.4. Importação de dados vetoriais no GRASS

Agora que já temos os shapefiles, vamos importá-los no GRASS. Vá em **File** → **Import vector data** → **Import of common formats with reprojection [v.import]**.

Na aba **Input**, podemos apenas selecionar o arquivo com extensão `.shp`. Na aba **Output**, se não inserirmos um nome novo para o layer, o nome do arquivo de entrada será utilizado. Ainda temos a opção de importar todo o arquivo (`extent-input`) ou limitar à `region` ativa (`extent-region`). Na aba **Input SRS** vamos inserir o código EPSG que anotamos no QGIS (lembra? era 4618). Como o arquivo é bem grande, vamos limitar a importação à `region` que salvamos anteriormente. Importe os três shapefiles.

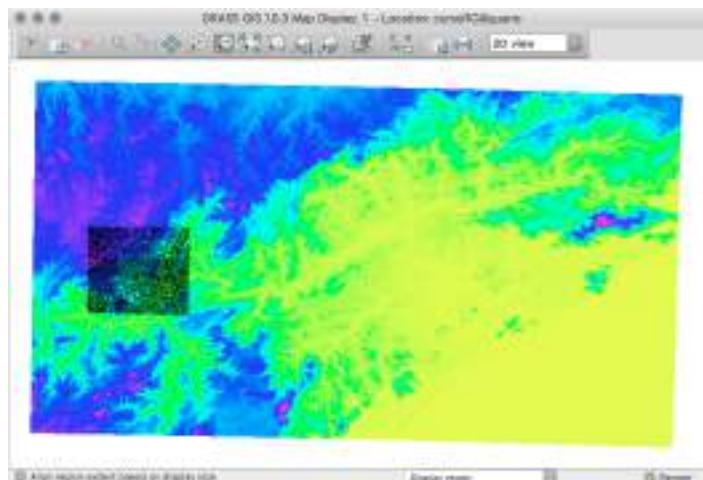


Figura 7.15. Dados vetoriais importados somente para a `region` ativa.

Um ponto interessante é que em um shapefile não é possível ter mais de um tipo de geometria (ponto, linha, polígono) por arquivo, mas o formato vetorial do GRASS não tem essa limitação. Quando um arquivo de polígonos é importado, temos as linhas que representam os limites dos polígonos (*boundaries*), as *áreas* dos polígonos e os *centróides* de cada área (algo como o centro de massa do polígono). Se quisermos, podemos pedir ao GRASS que exiba somente as linhas ou os centróides ou combinações das diversas geometrias.

O comando que usamos para importação é específico para casos em que o sistema de coordenadas do arquivo e da Location não são iguais. Quando os sistemas são iguais, podemos usar **File → Import vector data → Common import formats [v.in.ogr]**, que é uma versão simplificada do comando **v.in.ogr**. Se você rodar o comando **v.in.ogr** no terminal (sem nenhum parâmetro), pode ver que há muitas opções para a importação. Além de importar o arquivo para a **Location** ativa, pode-se importar apenas as entidades localizadas dentro de determinadas coordenadas (*subset*), ou entidades selecionadas pelos seus atributos, por meio de expressões SQL.

7.5. Buffers

Uma tarefa bastante comum em SIG é determinar áreas de vizinhança (*buffers*) ao redor de pontos, linhas ou áreas. No GRASS podemos fazer *buffers* para a partir de mapas *raster* ou vetoriais.

Vamos ver um exemplo. Vá em **vector → Buffer vectors [v.buffer]**. Primeiro use o mapa de drenagens que importamos como entrada e na aba **Distance** defina um valor de 50 (metros, a unidade das coordenadas). Assim todas as linhas terão um *buffer* com a mesma distância. Ainda na aba **Distance**, podemos definir a orientação de um eixo arbitrário e fazer *buffers* com distâncias diferentes nas direções paralela e perpendicular a esse eixo. Veja o efeito disso fazendo um *buffer* com distância de 200m no eixo principal (*major axis*), 400m no eixo perpendicular (*minor axis*) e direção do eixo principal de 45°(fig.7.16).

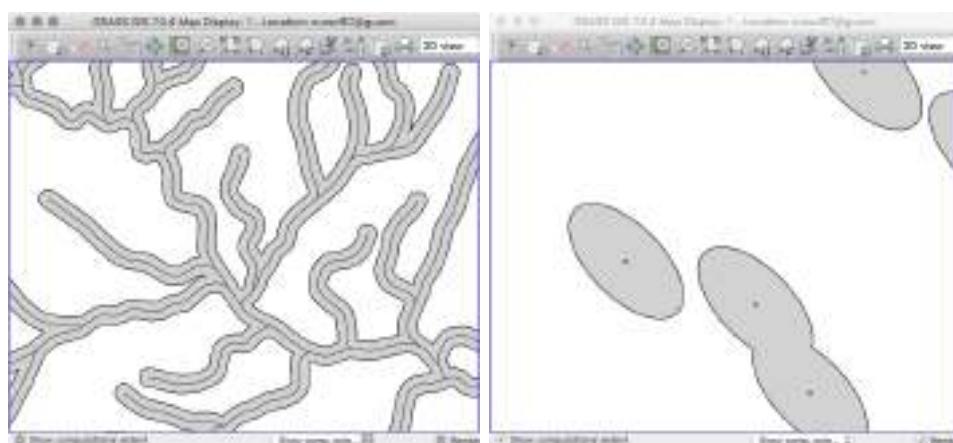


Figura 7.16. *Buffer* com distâncias fixas (esq.) e diferentes com relação a um eixo arbitrário (dir.).

Além de determinar um valor fixo para a distância do *buffer*, possível usar os valores de uma coluna do banco de dados para que categorias diferentes tenham distâncias diferentes. O mapa de **Pontos_Cotados** tem as altitudes dos pontos na coluna **AVG_Z**. Vamos fazer um *buffer* a partir desse mapa. Na aba **Selection**, escolha o *layer 1* e na aba **Distance** selecione a coluna **AVG_Z** (deixe as opções de distância em branco).

Veja que o raio de cada *buffer* varia de acordo com o valor no banco de dados e que os polígonos que se sobrepõem são fundidos. Caso não queira que isso aconteça, selecione a *flag Transfer categories and attributes* na aba **Attributes**. A diferença entre os resultados está na figura 7.17.

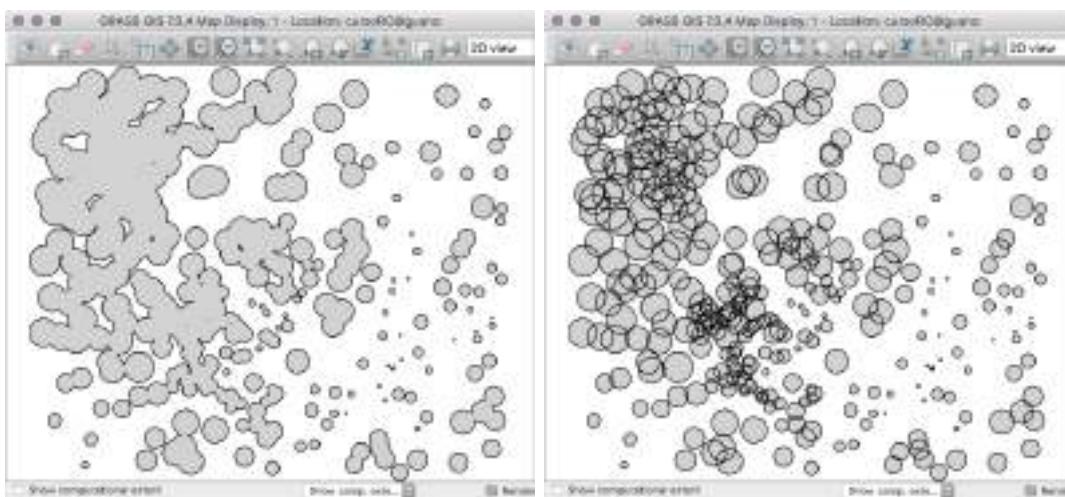


Figura 7.17. *Buffer* com fusão das geometrias (esquerda) e sem fusão (direita).

8. Trabalhando com máscaras

Máscaras são uma maneira de selecionar áreas irregulares para análises. Normalmente define-se uma área a partir de um polígono vettorial ou de uma (ou mais) classe de um mapa *raster*. Uma vez definida a máscara, as análises *raster* só serão executadas dentro da máscara, obedecendo as definições de extensão e resolução espacial da **region**.

Nossa máscara será definida pelos limites do Parque Estadual Turístico do Alto Ribeira (PETAR). Para obter o shapefile do PETAR, vá em <http://mapas.mma.gov.br/i3geo/datadownload.htm> e, no menu da esquerda em **i3Geo** → **Áreas Especiais** → **Unidades de conservação** → **UCs estaduais de proteção integral**. Clique sobre o **nome** do tema e baixe os três arquivos **ucsei**. Abra no QGIS (coordenadas geográficas, WHS84) e selecione o PETAR. Clique com o botão direito e escolha **Save As**. No diálogo de salvar, defina o sistema de coordenadas como **WGS84 UTM 22S** e lembre-se de marcar a opção **Save only selected features** (fig. 8.1).

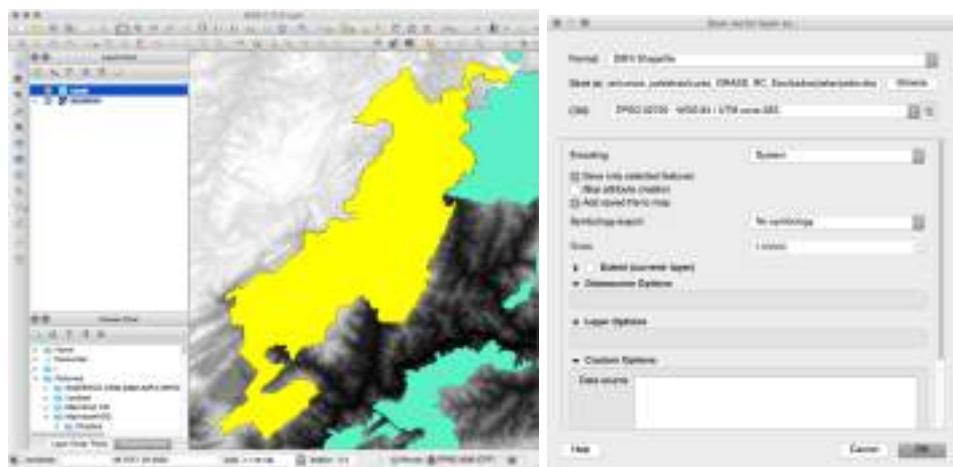


Figura 8.1. Salve apenas os limites do PETAR.

Importe o arquivo no GRASS e visualise o mapa sobre o modelo SRTM para ver se está tudo certo. Depois vá em **Raster** → **Mask [r.mask]**. Na aba **Vector**, defina o mapa do PETAR como máscara e rode o comando. Veja que agora só a porção do *raster* que está dentro dos limites do polígono é exibida (fig. 8.2).

Outra opção é criar uma máscara a partir de valores ou intervalos de valores de mapas *raster*. Remova a máscara existente (no terminal, use **r.mask -r**) e na aba **Raster**, selecione o mapa **srtm_30m** como entrada e em **maskcats** use um intervalo de valores, como **100 thru 300**. Você também pode usar mais de um intervalo, como em **100 thru 300 800 thru 1200** (fig. 8.3).

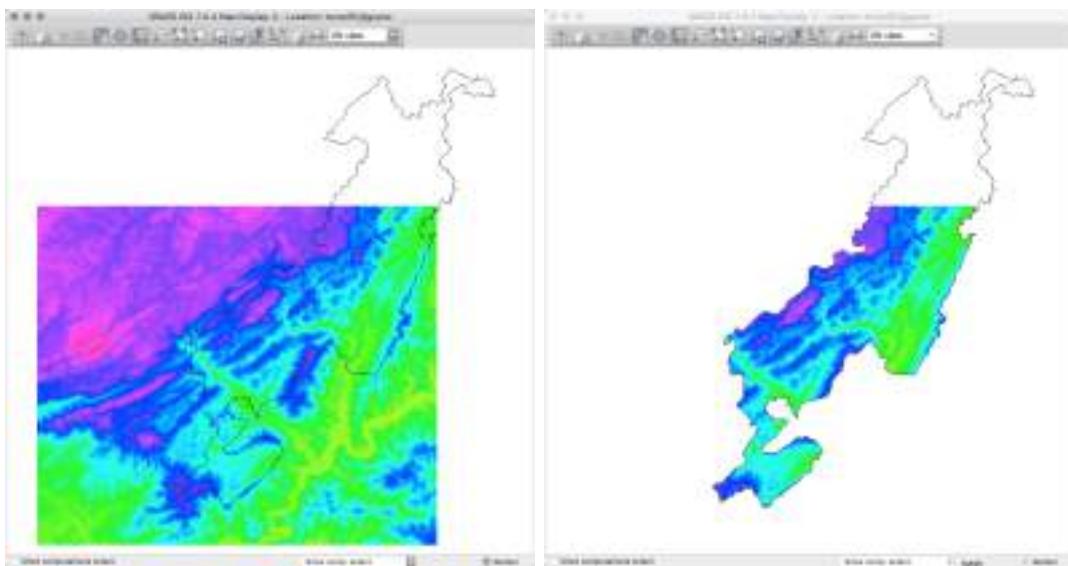


Figura 8.2. Efeito da máscara sobre dados *raster*.

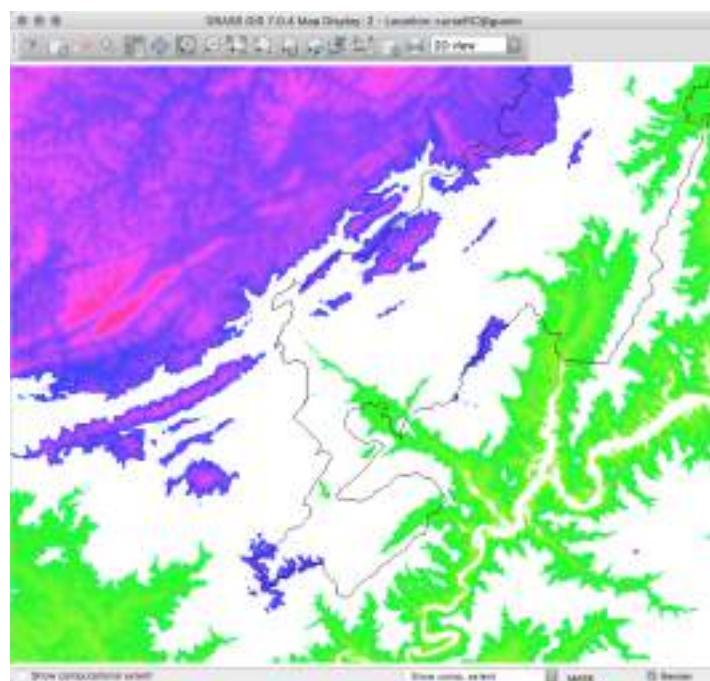


Figura 8.3. Máscara com dois intervalos de valores (100 thru 300 800 thru 1200).

9. Análise Digital de Terreno

9.1. Hipsometria

Vamos iniciar a análise de terreno com o mapa hipsométrico, onde temos a representação do relevo em classes de altitude (no nosso caso, classes de 50 metros). Vamos utilizar os comandos `r.reclass` para criar o mapa com as classes de altitude e o comando `r.colors` para definir a cor de cada classe.

Vá em **Raster → Change category values and labels → Reclassify [r.reclass]**. Na aba **Required**, especifique o nome do mapa a ser reclassificado (no nosso caso, `srtm_30m`) e o nome do novo mapa a ser criado, como `srtm_30m_hipso100`. Ainda podemos indicar um arquivo de texto que contenha as regras de classificação ou entrar com os valores manualmente.

As regras de classificação devem ser digitadas de acordo com o seguinte esquema:

intervalo de dados = número da categoria rótulo da categoria

No nosso exemplo, usamos as seguintes regras:

```
-50 thru 100 = 1 0 - 100m
100 thru 200 = 2 100 - 200m
200 thru 300 = 3 200 - 300m
300 thru 400 = 4 300 - 400m
400 thru 500 = 5 400 - 500m
500 thru 600 = 6 500 - 600m
600 thru 700 = 7 600 - 700m
700 thru 800 = 8 700 - 800m
800 thru 900 = 9 800 - 900m
900 thru 1000 = 10 900 - 1000m
1000 thru 1500 = 11 1000 - 1500m
```

Note que um intervalo dos dados é definido por *menor valor thru maior valor*. Outro ponto importante é que o comando `r.reclass` na verdade **não** gera um layer novo, mas apenas uma tabela de reclassificação que é interpretada pelo GRASS cada vez que é necessário acessar o layer novo (reclassificado). Isto é feito para economizar espaço em disco e faz com que, enquanto o layer reclassificado existir, não seja possível apagar ou renomear o layer original, pois a reclassificação depende dele para ser acessada.

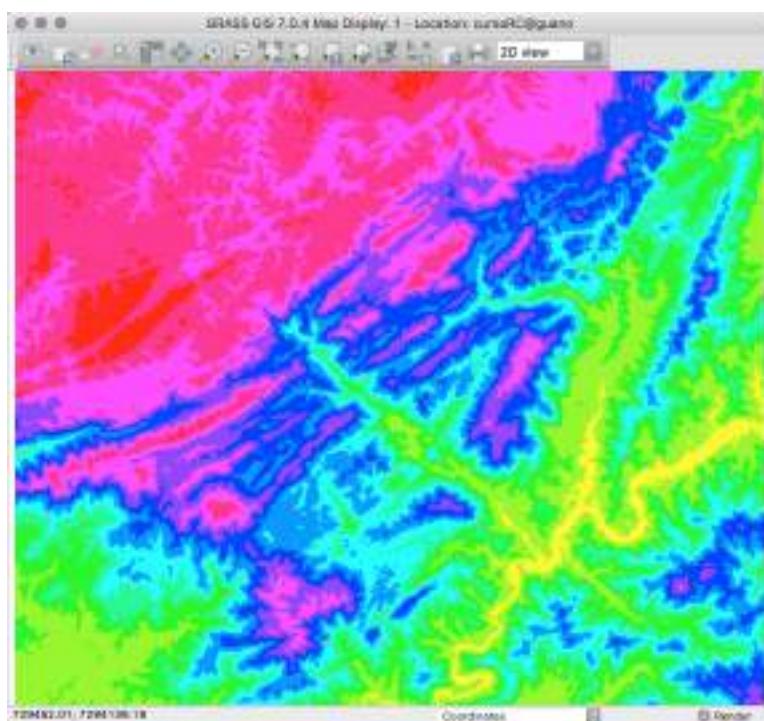


Figura 9.1. Reclassificação dos dados SRTM em classes de 100 metros.

Para alterar as cores do mapa reclassificado, vamos usar o comando `r.colors` (**Raster** → **Manage colors** → **Color tables [r.colors]**). Podemos escolher esquemas pré-definidos ou utilizar regras próprias. O comando `r.colors` oferece várias tabelas de cores pré-definidas (fig. 9.2).



Figura 9.2. Escalas de cores do comando `r.colors`.

As regras de classificação devem ser digitadas da seguinte maneira:

número da categoria	valor_RED	valor_GREEN	valor_BLUE
---------------------	-----------	-------------	------------

Onde os valores para as cores estão entre 0-255. Existem outras opções para definir cores com `r.colors`, como utilizar porcentagens de Vermelho/Verde/Azul (RGB) ao invés de valores entre 0-255, utilizar nomes de cores ou esquemas pré-definidos. Para ver todas as possibilidades, consulte a página de manual do comando.

Na figura 9.3 temos o mapa hipsométrico com a tabela de cores `elevation`.

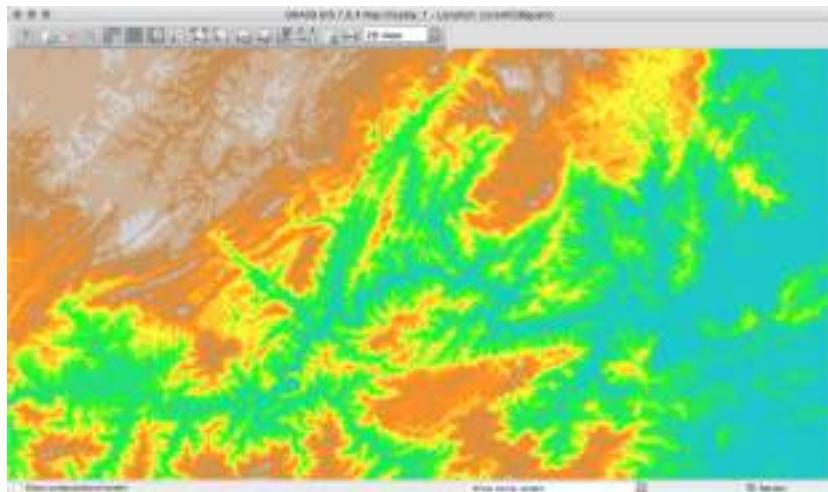


Figura 9.3. Mapa hipsométrico com classes de 100 metros e tabela de cores `elevation`.

9.2. Relevo sombreado

Vamos fazer uma imagem de relevo sombreado (*shaded relief*) dos nossos dados SRTM. Este comando vai simular uma iluminação solar sobre o modelo de elevação. Para isso precisamos informar a **altitude** do sol com relação ao horizonte (no exemplo, 25°) e também o seu **azimute** (45°). Adicionalmente, podemos utilizar um **exagero vertical** (5), muito útil em áreas sem desnível acentuado.

Vá em **Raster → Terrain analysis → Compute shaded relief [r.relief]**. Defina a **region** para coincidir com os limites e resolução do raster `srtm_30m`.

Na aba **Required**, entramos com o mapa `srtm_30m` e um nome para o mapa de saída. As opções da posição do sol estão na aba **Sun position**, e na aba **Optional** temos o exagero vertical e um fator de correção para unidades de elevação. Esse comando espera que as unidades dos dados na horizontal e na vertical sejam as mesmas (isso vale para outros SIGs). Ao trabalhar com Locations em latitude-longitude, por exemplo, as unidades horizontais estarão em graus e as verticais em metros. Nesse caso, deveríamos selecionar **meters** na última opção, que faz o ajuste entre as escalas horizontal e vertical.

Para não perder o costume de usar nomes explicativos, uma boa opção para o nome do mapa de saída é `srtm_30m_shade_045_25_z2`. Isso significa que o mapa base foi o `srtm_30m`, que foi aplicado um sombreamento `shade`, com azimute de 045 graus, inclinação de 25 graus e exagero vertical (em `z`) de 2 vezes. É sempre bom utilizar três algarismos para representar o azimute, para evitar confusão com a inclinação. O resultado está na figura (9.4).

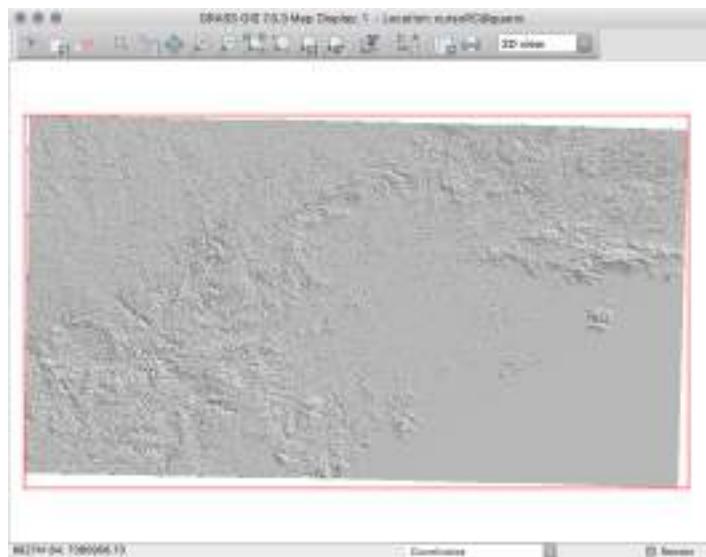


Figura 9.4. Resultado do comando `r.relief`, com resolução original de 30m.

Não dá para ver muita coisa. Isso porque nossa área é muito grande, e a resolução original dos dados (3 arcos de segundo, ou aproximadamente 90 m) é muito detalhada para observar a área como um todo. Para poder ver as formas de relevo maiores, precisamos diminuir a resolução espacial.

Vamos alterar a resolução espacial para 300m com `g.region -p res=300`. Agora refaça a imagem de relevo sombreado, usando os mesmos parâmetros de iluminação e exagero vertical. Na figura 9.5 podemos ver bem melhor o relevo da região.

Uma opção interessante é visualizar o relevo sombreado com a escala de cores sobreposta. Para isso, no *Layer Manager*, clique no ícone **Add various raster map layers** e escolha **Add shaded relief map layer**. Na janela do comando `d.shade`, selecione `srtm_300m_shade_045_25_z2` como imagem de relevo (`shade`) e `srtm_30m_hipso` como imagem a ser sobreposta ao relevo (`color`). Visualize o resultado no *Map Display* (fig. 9.6).

A aparência da imagem produzida varia bastante de acordo com o azimute do iluminante. Feições lineares do relevo (que podem estar ligadas à estruturas geológicas, por exemplo) são destacadas quando o iluminante está posicionado na direção perpendicular à feição. Caso o interesse seja uma análise de elementos lineares interpretados em imagens de relevo sombreado, deve-se produzir vários mapas, com iluminante posicionado em diferentes orientações (Liu, 1987; Riccomini & Crósta, 1988; Grohmann, 2004; Smith & Wise, 2007).

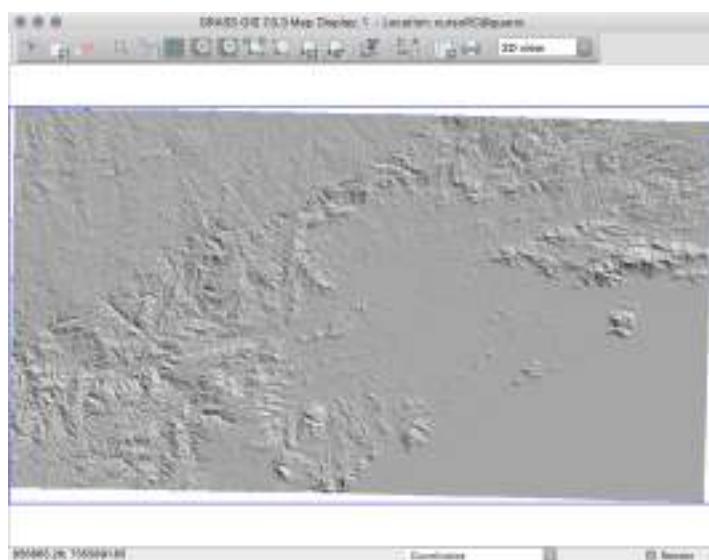


Figura 9.5. Resultado do comando `r.relief`, com resolução de 300m.

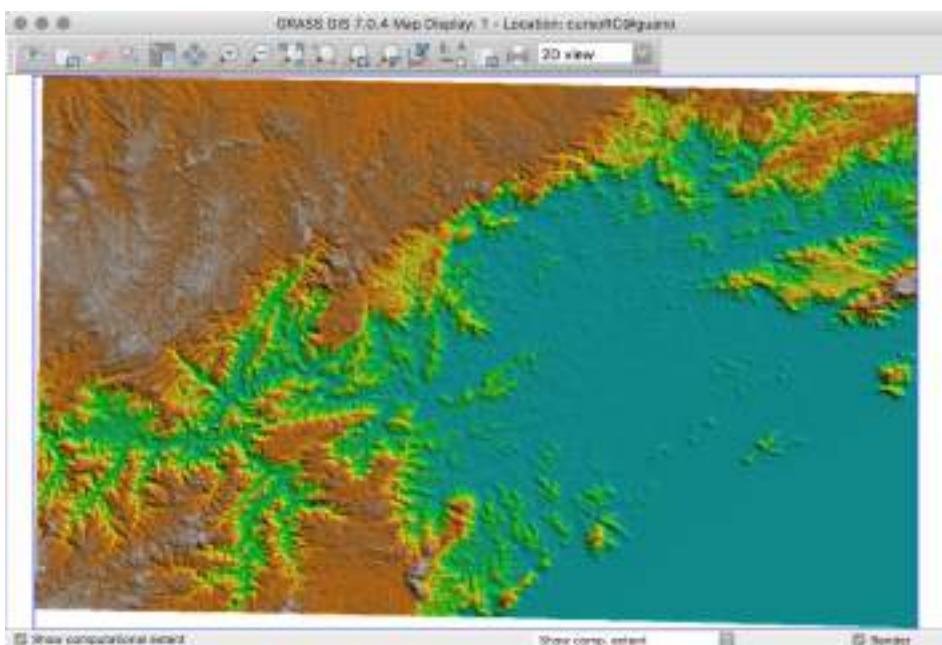


Figura 9.6. Mapa hipsométrico sobreposto a relevo sombreado.

A posição do iluminante pode até mesmo fazer com que tenhamos a impressão de “relevo invertido”. Pessoas que moram no hemisfério sul, por exemplo, estão acostumadas com iluminação vinda de norte em aerofotografias e em imagens orbitais. Um mapa de relevo sombreado com iluminação vinda de sul pode parecer invertido para alguns.

9.3. Índices de forma

Diversos parâmetros morfométricos podem ser usados para descrever a superfície topográfica, tais como declividade da vertente (*slope*), orientação da vertente (azimute ou aspecto, *aspect*) e curvaturas associadas à forma concáva/convexa da forma de relevo.

Para calcular esses índices, vamos usar o comando **r.slope.aspect** (**Raster → Terrain analysis → Slope and aspect [r.slope.aspect]**). Nossa mapa de entrada é **srtm_30m**. Na aba **Output**, defina os nomes dos mapas de saída, de acordo com o parâmetro calculado: **srtm_30m_slope** e **srtm_30m_aspect**. Não se esqueça de conferir as configurações da **region** ativa antes de executar o comando.

Por padrão o mapa de declividade tem valores em graus, ou opcionalmente em porcentagem. O mapa de orientação de vertentes tem valores iniciando em Leste, crescendo em sentido anti-horário: 90° corresponde a Norte, 180° a Oeste, 270° a Sul e 360° a Leste. O valor 0 (zero) é usado para identificar áreas de orientação indefinida quando a superfície é plana (declividade zero). Caso a unidade dos valores de elevação do layer original não esteja em metros, é preciso usar a opção **zscale** para converter os valores.

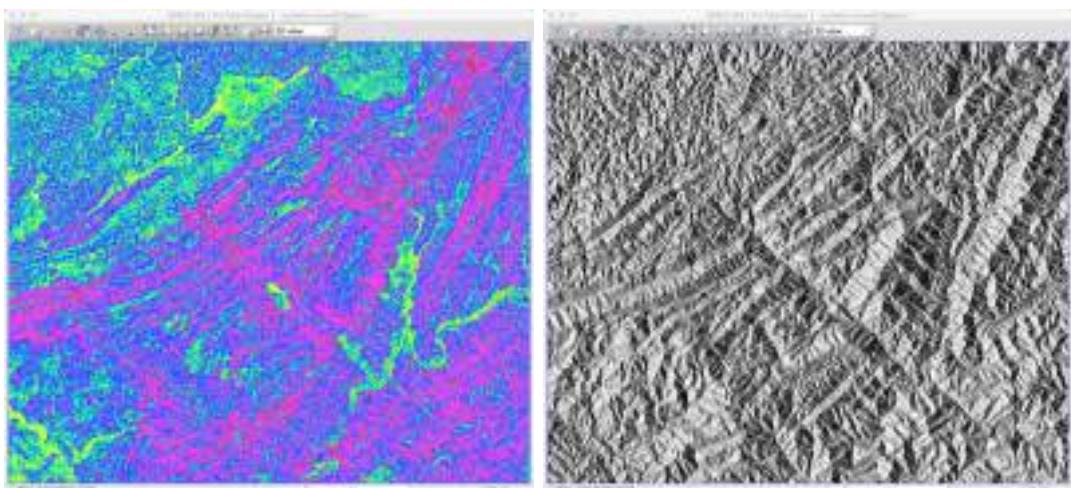


Figura 9.7. Mapas de declividade (esquerda) e aspecto (orientação de vertentes - direita).

Dependendo da aplicação, o usuário pode necessitar de uma mapa com declividade agrupadas em classes. Isto pode ser feito com o comando **r.reclass**. Um exemplo de classificação é dado na página de manual do comando **r.slope.aspect**:

categoria	intervalo (em graus)	rótulos (labels) (em porcentagem)
1	0-1	0-2%
2	2-3	3-5%
3	4-5	6-10%
4	6-8	11-15%
5	9-11	16-20%
6	12-14	21-25%
7	15-90	26% e acima

A regra de reclassificação do `r.reclass` ficaria assim:

```
0 thru 1      = 1  0 thru 2%
2 thru 3      = 2  3 thru 5%
4 thru 5      = 3  6 thru 10%
6 thru 8      = 4  11 thru 15%
9 thru 11     = 5  16 thru 20%
12 thru 14    = 6  21 thru 25%
15 thru 90    = 7  26% e acima
```

Os cálculos deste comando são realizados de acordo com as fórmulas de Horn (1981). Para a determinação da declividade e do aspecto, o algoritmo utiliza uma matriz 3x3 ao redor de cada pixel, portanto não é possível calcular os valores para as células adjacentes às bordas do layer. Essas células recebem valores de declividade e aspecto zero.

De acordo com o manual do comando, caso o layer utilizado para calcular o aspecto seja do tipo **integer** (apenas valores inteiros), deve ocorrer maior freqüência de valores nas direções 0° , 45° , 90° , 180° , 225° , 270° e 315° . Caso o layer seja do tipo **float** (valores fracionais), esse tipo de problema não deve acontecer. Veremos como resolver esse problema na seção 10.6.

Células com declividade muito baixa também terão maior freqüência de valores nas direções 0° , 45° , 90° , 180° , 225° , 270° e 315° . Para evitar que isso ocorra, pode-se utilizar a opção **min_slp**, que fará com que todas as células com declividade menor que o valor estabelecido recebam valores nulos.

O mapa de aspecto produzido pelo GRASS tem valores entre 0-360 iniciando no Leste e crescendo em sentido anti-horário. Essa organização dos valores é esperada por alguns outros comandos (como os de modelagem hidrológica), e enquanto que a escala de cores utilizada é muito boa para interpretações visuais, se quisermos fazer uma análise estatística ou correlação com o mundo real, devemos transformar os valores para a orientação geográfica (*compass-oriented aspect values*).

Podemos fazer isso com a ferramenta de álgebra de mapas raster, `r.mapcalc`. Vá em **Raster → Raster map calculator [r.mapcalc]**. Aqui entramos com uma expressão que será avaliada pelo comando e executada, para um ou mais rasters.

Pode-se utilizar desde expressões simples até esquemas complexos, envolvendo testes lógicos. Uma boa leitura sobre essa poderosa ferramenta é Shapiro & Westervelt (1991).

No campo maior da parte inferior da janela, **Expression**, podemos digitar as expressões ou podemos utilizar a segunda opção da parte direita da janela, **Insert mapcalc function** para nos auxiliar. Procure por `if(x,a,b)` e a insira. Essa expressão irá primeiro verificar a condição `x`. Se `x` for verdadeiro, então a instrução `a` será executada. Se `x` for falso, a instrução `b` será executada.

Para converter os valores de aspecto, usaremos a seguinte fórmula:

```
if(A <= 90, 90-A, 360+90-A)
```

Portanto se os valores do mapa `A` forem menores ou iguais a 90, iremos subtrai-los de 90, caso os valores sejam maiores do que 90, eles serão subtraídos de 450.

Preencha os campos da janela (fig: 9.8) de modo que o mapa de saída seja `srtm_30m_aspect_compass` e a expressão seja:

```
if(srtm_30m_aspect <= 90, 90 - srtm_30m_aspect, 450 - srtm_30m_aspect)
```

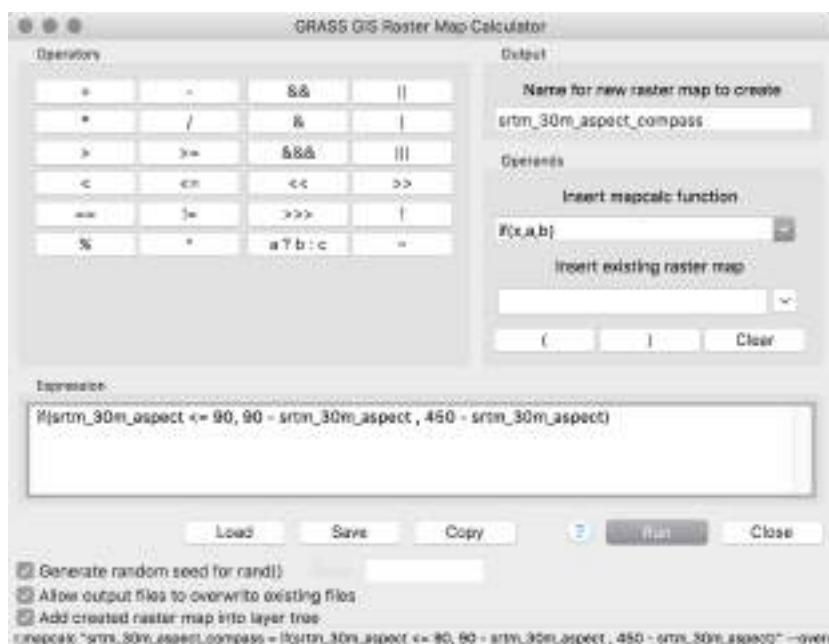


Figura 9.8. Janela do `r.mapcalculator`.

Após a execução do `r.mapcalc`, o GRASS atribui ao novo mapa uma escala de cores padrão, mas você pode mudar isso usando `r.colors` e atribuir, por exemplo, a escala `aspect` ou `aspectcolr`. Se fizer isso, irá notar que a escala de cores `aspect` deixa o mapa um pouco estranho, até com aparência de relevo invertido (fig: 9.9). Como falamos na seção 9.2, estamos acostumados a imagens com iluminante a partir do norte, então podemos acertar essa escala de cores usando o seguinte esquema em `r.colors`:

0 white
180 black
360 white

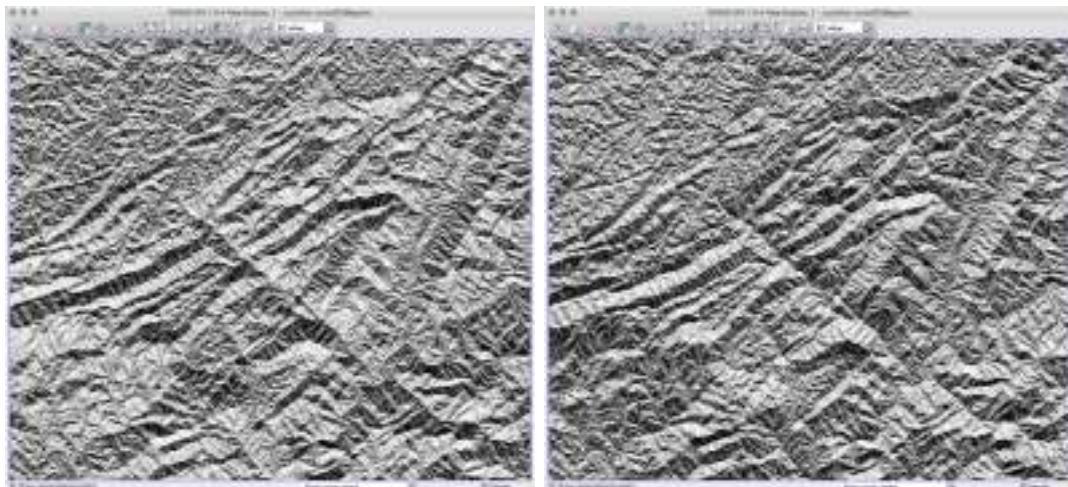


Figura 9.9. Mapas aspecto com escala de cores aspect (esquerda) e definida manualmente (direita).

9.4. Análise de vizinhança

A análise de vizinhança baseia nas relações entre os valores das células de um *raster* com as células vizinhas e podem ser calculados por janelas móveis ou por janelas de busca.

As *janelas móveis* (*roving-windows* – Demers, 2004), também chamadas de médias móveis ou filtros de convolução (este termo mais utilizado em processamento digital de imagens), consiste em estabelecer uma vizinhança de $(n \times n)$ células (com n ímpar, geralmente) ao redor de cada célula do mapa *raster*, e estabelecer a esta célula central um valor determinado a partir de uma função (média, moda, desvio padrão etc) de todos os valores contidos na vizinhança (fig. 9.10).

Na análise por *janelas de busca* (*search window* – Shary *et al.*, 2002) estabelece-se uma janela (normalmente quadrada) de tamanho arbitrário, a função é calculada para todas as células que estiverem dentro da janela e o valor obtido é atribuído ao ponto central da janela (de acordo com suas coordenadas, não necessariamente coincidente com uma célula do mapa). A janela é então deslocada de acordo com sua extensão E-W ou N-S, de modo a perfazer uma série de “linhas” e “colunas” contíguas. Os valores obtidos são então interpolados a fim de obter uma representação contínua da variável em questão (fig. 9.10).

Dependendo da relação de escala entre o tamanho da vizinhança utilizada e a morfologia da superfície estudada, a análise por janelas móveis pode gerar uma representação de valores extremos locais (máximos ou mínimos) como “platôs” (vistos no perfil esquemático da figura 9.11B).

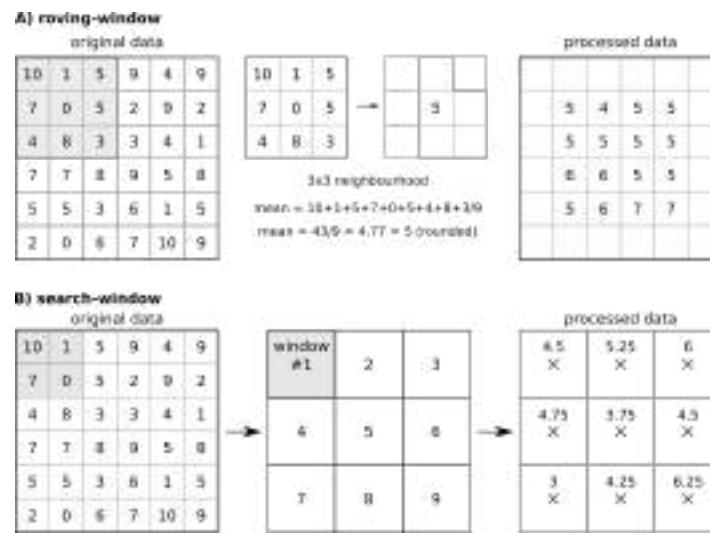


Figura 9.10. Análise por janelas móveis (A) e por janela de busca (B) (Grohmann & Riccomini, 2009).

Ao se trabalhar com janelas móveis, todas as células do mapa são consideradas (exceto aquelas localizadas nas bordas do mapa, que não têm seus valores calculados por não possuírem uma vizinhança completa), portanto não há risco de que valores importantes (como máximos e mínimos locais) sejam “perdidos” ou “deslocados”, como pode ocorrer com janelas de busca, dependendo do tamanho da janela utilizada e sua posição (fig. 9.11C e D).

O custo computacional envolvido é diretamente proporcional ao tamanho da vizinhança e, em menor grau, à operação realizada. Vizinhanças menores são processadas rapidamente ao passo que para vizinhanças maiores a quantidade de valores envolvidos aumenta exponencialmente. Para o método de janelas de busca, a situação se inverte. Quanto maior a janela, menor o número de vezes que a operação deverá ser repetida e a interpolação demandará menos tempo, enquanto que com janelas pequenas, o número de operações se torna elevado, e a interpolação irá consumir mais recursos do sistema.

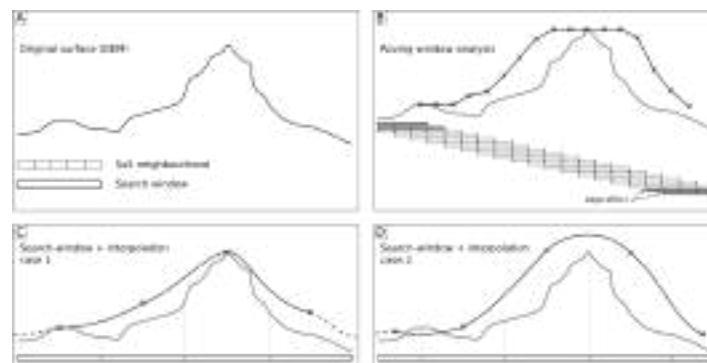


Figura 9.11. Comparação entre análise por janelas móveis e por janela de busca (opération de máximo). A) Superfície (modelo de elevação) original; B) Análise por janelas móveis (5x5); C) Análise por janela de busca – caso 1; D) Análise por janela de busca – caso 2 (Grohmann & Riccomini, 2009).

No GRASS temos o comando `r.neighbors` para realizar as operações de janelas móveis. As janelas devem ter número ímpar de pixels em cada lado e podem ser quadradas ou circulares. As operações disponíveis são:

`average` - média dos valores na janela, arredondada para o inteiro mais próximo;
`median` - o valor a 'meio caminho' da lista ordenada dos valores da janela;
`mode` - o valor que ocorre mais frequentemente na janela;
`minimum` - o menor dos valores na janela;
`maximum` - o maior dos valores na janela;
`range` - o intervalo dos valores na janela;
`stddev` - o desvio padrão dos valores na janela, arredondado para o inteiro mais próximo;
`sum` - a soma dos valores na janela;
`variance` - a variância dos valores na janela, arredondada para o inteiro mais próximo;
`diversity` - o número de valores diferentes na janela;
`interspersion` - a porcentagem de células com valores diferentes do valor do pixel central, mais 1; arredondada para o inteiro mais próximo.

Vamos ver isso na prática. O desvio padrão da declividade tem sido utilizado por alguns autores como uma medida da rugosidade do relevo (Grohmann *et al.*, 2010). Cheque as definições da `region` (pra não perder o costume) e vá em **Raster → Neighborhood analysis → Moving windows [r.neighbors]**. Como mapa de `input`, use o mapa de declividade `srtm_30m_slope`. Na aba **Neighborhood** escolha a opção `stddev` e defina um tamanho de janela (`5x5 pixels`, por exemplo). Dê um nome decente para o mapa de saída (`petar_topografia_rst_slope_sttdev_5x5`). Faça outro com janela de `21x21 pixels` e compare os dois usando a visualização sobre um relevo sombreado (fig. 9.12)

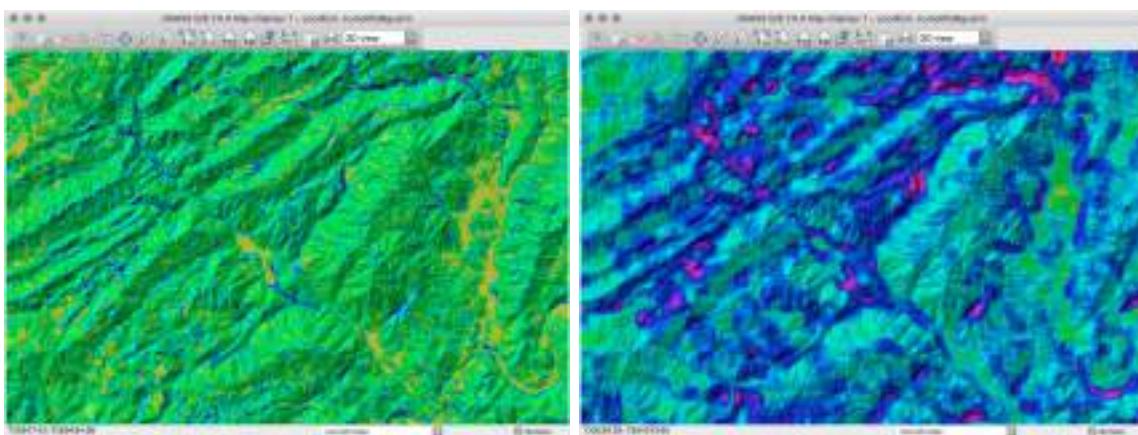


Figura 9.12. Comparação de operações de janelas de busca (`stddev`) com `5x5` (esquerda) e `21x21 pixels` (direita).

9.5. Parâmetros do relevo

Um comando interessante na análise de formas de relevo é o `r.param.scale`, que calcula diversos parâmetros relacionados à superfície topográfica a partir de janelas móveis. Os cálculos são feitos ao se ajustar uma superfície quadrática aos dados da janela, pelo método dos mínimos quadrados (Wood, 1996a,b, 2009).

Além de calcular parâmetros como declividade, aspecto e curvaturas, podemos utilizar o comando para reamostrar a topografia de acordo com o tamanho da janela e ainda classificar as formas do relevo de acordo com sua morfologia (fig. 9.13).

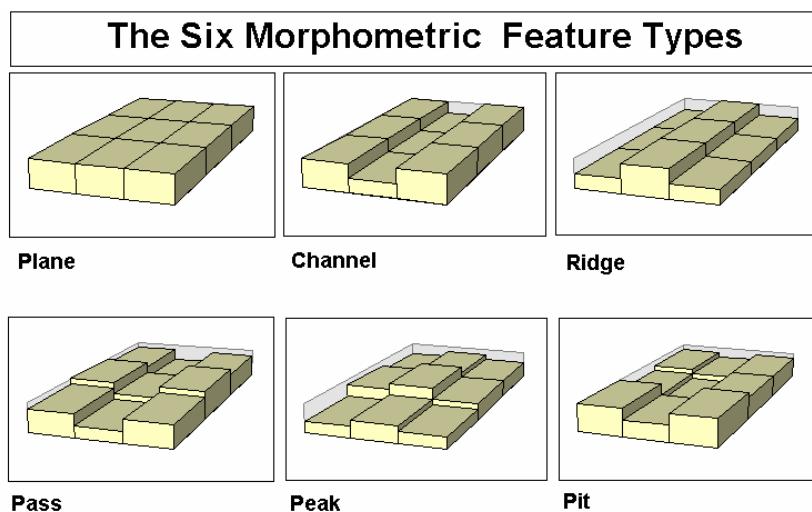


Figura 9.13. Feições morfométricas do relevo (Wood, 1996b)

Vamos ver como o comando se comporta na nossa área de trabalho. Vá em **Raster** → **Terrain analysis** → **Terrain Paarameters [r.param.scale]**. Confira as definições da `region` e veja se não há nenhuma máscara ativa. Nosso mapa de entrada pode ser o `srtm_petar_25m_bilinear` e o mapa de saída, `srtm_petar_25m_bilinear_feature`.

Na aba **Optional** temos as configurações do comando. Aqui definimos o tamanho da janela em *pixels* (deve ser ímpar), o parâmetro a ser calculado (escolha `feature`) e alguns valores que definem como o comando vai determinar a geometria da superfície. A tolerância da curvatura (`curvature_tolerance`) influencia muito no resultado final. Faça um mapa com janela de 21 *pixels* e tolerância de curvatura 0.0003. Depois faça outro com a mesma janela mas tolerância de 0.0008 e compare os resultados (fig. 9.14).

As categorias geradas são: 1 - Planar, 2 - Pit, 3 - Channel, 4 - Pass (saddle), 5 - Ridge, 6 - Peak. Note o “efeito de borda” do método de janelas móveis nos limites do mapa. Os pixels das bordas são perdidos por não possuírem uma vizinhança completa.

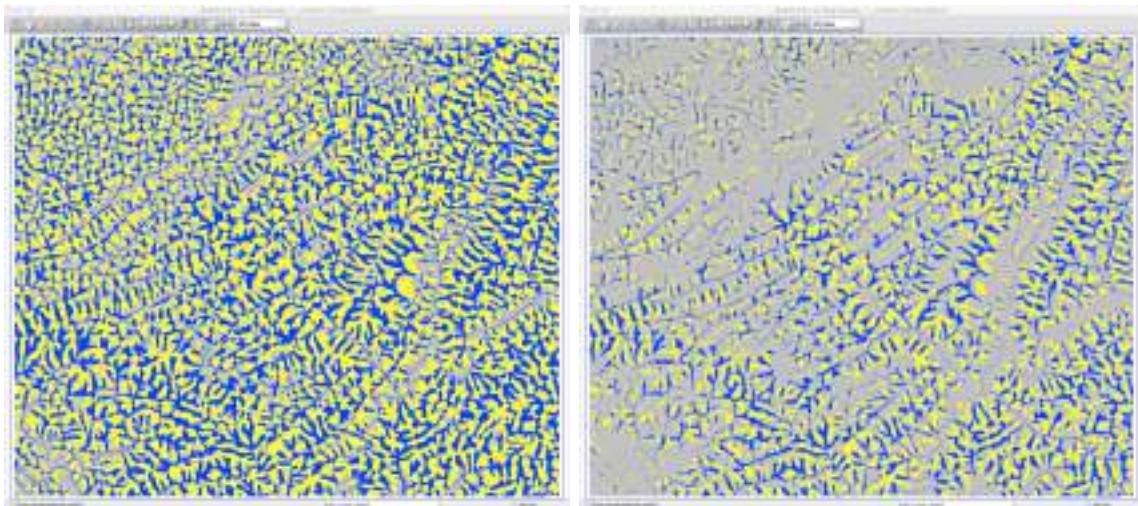


Figura 9.14. Resultado do comando `r.param.scale` com janela de 21 *pixels* e tolerância de curvatura 0.0003 (esquerda) e 0.0008 (direita).

Vamos ver essa diferença no resultado com estatísticas para cada mapa usando o comando `r.univar` (**Raster → Reports and statistics → Univariate raster statistics** [`r.univar`]). Para gerar as estatísticas para cada categoria, vamos usar máscaras. Para cada mapa, vamos definir a máscara pelo valor da categoria (de 1 a 6) e rodar o `r.univar`. O exemplo abaixo já mostra como podemos pensar em automatizar as tarefas. Podemos escrever esses comandos em um editor de texto simples (como o Notepad) e depois simplesmente copiar e colar os comandos no terminal. As linhas que começam com sinal de *hash* (#) são ignoradas e as usamos como comentários para nos ajudar a lembrar o que estamos fazendo e organizar melhor as coisas.

```
# mapa srtm_petar_25m_bilinear_feature_21_0003
# categoria 1 (planar)
r.mask --overwrite raster=srtm_petar_25m_bilinear_feature_21_0003@guano maskcats=1
r.univar map=srtm_petar_25m_bilinear@guano

# categoria 2 (pit)
r.mask --overwrite raster=srtm_petar_25m_bilinear_feature_21_0003@guano maskcats=2
r.univar map=srtm_petar_25m_bilinear@guano

# categoria 3 (channel)
r.mask --overwrite raster=srtm_petar_25m_bilinear_feature_21_0003@guano maskcats=3
r.univar map=srtm_petar_25m_bilinear@guano

# categoria 4 (pass)
r.mask --overwrite raster=srtm_petar_25m_bilinear_feature_21_0003@guano maskcats=4
r.univar map=srtm_petar_25m_bilinear@guano
...
```

Aqui você vê as diferenças entre as estatísticas para a classe 1 (planar), dos mapas gerados com tolerância de curvatura 0.0003 (esquerda) e 0.0008 (direita):

```
r.univar map=srtm_petar_25m_bilinear@guano      r.univar map=srtm_petar_25m_bilinear@guano
total null and non-null cells: 1792560          total null and non-null cells: 1792560
total null cells: 1185455                      total null cells: 561069
Of the non-null cells:                           Of the non-null cells:
-----
n: 607105                                         n: 1231491
minimum: 27.9097                                    minimum: 27.9097
maximum: 1180.44                                    maximum: 1189.58
range: 1152.53                                     range: 1161.67
mean: 598.48                                       mean: 596.242
mean of absolute values: 598.48                   mean of absolute values: 596.242
standard deviation: 305.195                        standard deviation: 296.616
variance: 93144                                     variance: 87981
variation coefficient: 50.995 %                   variation coefficient: 49.7476 %
sum: 363340433.666669                          sum: 734266128.965282
```

Outra opção é definir um mapa com categorias para o `r.univar` usando a opção **zones**. Para nosso exemplo das classes de parâmetros do relevo, ficaria assim:

```
r.univar map=srtm_petar_25m_bilinear zones=srtm_petar_25m_bilinear_feature_21_0003

zone 1 Planar
total null and non-null cells: 1214210
total null cells: 0
Of the non-null cells:
-----
n: 1214210
minimum: 1
maximum: 1180.44
range: 1179.44
mean: 299.74
mean of absolute values: 299.74
standard deviation: 368.535
variance: 135818
variation coefficient: 122.951 %
sum: 363947538.66667

zone 2 Pit
total null and non-null cells: 1912
total null cells: 0
Of the non-null cells:
-----
```

```
n: 1912
minimum: 2
maximum: 1014.42
range: 1012.42
mean: 316.075
mean of absolute values: 316.075
standard deviation: 381.436
variance: 145493
variation coefficient: 120.679 %
sum: 604335.75
```

```
zone 3 Channel
total null and non-null cells: 1023380
total null cells: 0
Of the non-null cells:
-----
n: 1023380
minimum: 3
maximum: 1171.31
range: 1168.31
mean: 264.51
mean of absolute values: 264.51
standard deviation: 322.564
variance: 104048
variation coefficient: 121.948 %
sum: 270694098.173617
```

```
zone 4 Pass (saddle)
total null and non-null cells: 8028
total null cells: 0
Of the non-null cells:
-----
n: 8028
minimum: 4
maximum: 1164.17
range: 1160.17
mean: 363.052
mean of absolute values: 363.052
standard deviation: 409.034
variance: 167309
variation coefficient: 112.665 %
sum: 2914581.9375
```

```
zone 5 Ridge
total null and non-null cells: 1228510
total null cells: 0
Of the non-null cells:
-----
n: 1228510
minimum: 5
maximum: 1190.17
range: 1185.17
mean: 300.618
mean of absolute values: 300.618
standard deviation: 351.827
variance: 123782
variation coefficient: 117.034 %
sum: 369312437.222229
```

```
zone 6 Peak
total null and non-null cells: 2400
total null cells: 0
Of the non-null cells:
-----
n: 2400
minimum: 6
maximum: 1183.69
range: 1177.69
mean: 407.98
mean of absolute values: 407.98
standard deviation: 436.476
variance: 190512
variation coefficient: 106.985 %
sum: 979152.70138889
```

10. Interpolação de superfícies

Uma das tarefas mais comuns em SIGs é a interpolação de valores pontuais (distribuídos regular ou irregularmente no espaço) em uma superfície contínua, que pode ser usada em diversas modelagens e simulações. Provavelmente o exemplo mais comum seja a geração de Modelos Digitais de Elevação (MDEs), a partir de valores pontuais de elevação ou curvas de nível digitalizadas de mapas topográficos.

Entre as aplicações principais para interpolação estão:

- Alterar a resolução espacial de dados raster (**reamostragem**)

Métodos dos vizinhos mais próximos (*Nearest neighbor - NN*)

Interpolação por Splines (*Regularized Splines with Tension - RST*)

- Preenchimento de áreas sem dados (**interpolação**)

Inverso da potência da distância (*Inverse Distance Weighted - IDW*)

Interpolação por Splines (*Regularized Splines with Tension - RST*)

10.1. Reamostragem

A reamostragem de mapas raster pode ser feita de várias maneiras, como por vizinhos mais próximos (módulo `r.resample`), por *Splines* (`r.resamp.rst`) ou outros métodos de interpolação (`r.resamp.interp`). No primeiro caso, o mapa resultante terá a resolução da região ativa, e o valor de cada pixel será dado pelo valor do pixel original que estiver mais próximo do centro do novo pixel (figs. 10.1 e 10.2). Nos demais casos, pode-se ajustar diversos parâmetros da interpolação para um melhor resultado.

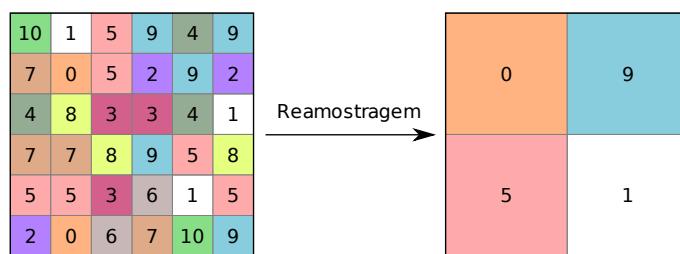


Figura 10.1. Reamostragem (mudança na resolução espacial) por vizinhos mais próximos.

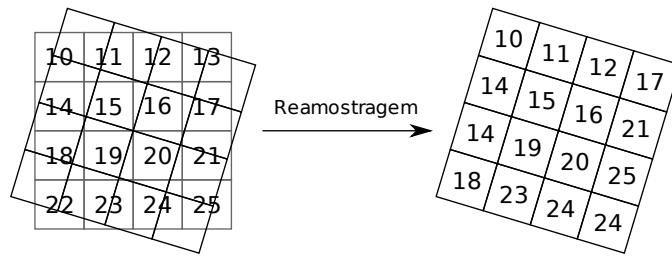


Figura 10.2. Reamostragem (transformação geométrica) por vizinhos mais próximos.

10.2. Inverso da potência da distância - IDW

Trata-se de um método de interpolação local, que se baseia na premissa de que pontos próximos possuem valores mais similares que pontos distantes. A variação entre os valores é modelada segundo potências (quadrado ou cubo) do inverso da distância entre os pontos (Fig. 10.3).

Módulos do GRASS para interpolação por IDW (o resultado será sempre um mapa raster):

- **v.surf.idw** – dados vetoriais (pontos, curvas de nível vetoriais)
- **r.surf.idw** – dados raster

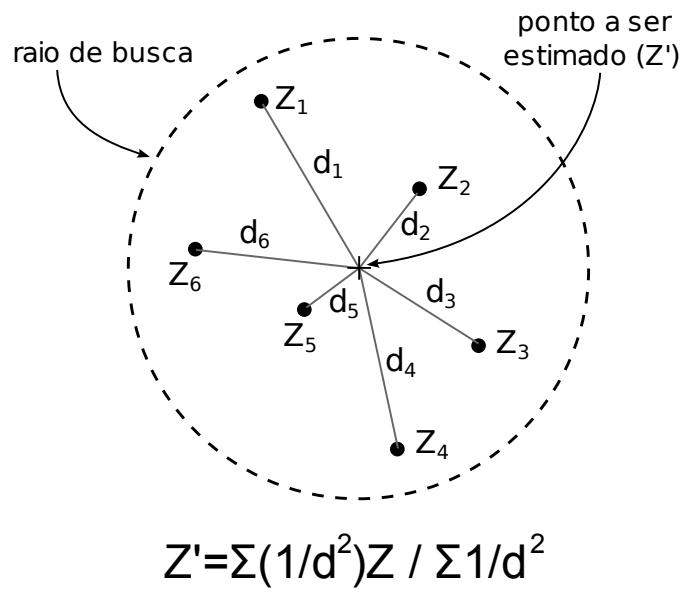


Figura 10.3. Esquema da interpolação por IDW.

10.3. Splines regularizadas com tensão - RST

O método de interpolação/aproximação por splines também é apropriado para o preenchimento de “vazios” nos dados (como a área de um lago, por exemplo) A idéia é gerar uma superfície que passe exatamente pelos pontos de dados ou próximo deles o suficiente para gerar uma superfície contínua e que preencha os espaços vazios (*data blanks, voids*). O ajuste dos parâmetros de interpolação muda o comportamento da superfície calculada, desde uma membrana elástica passando exatamente por todos os pontos até uma superfície “mais rígida”, que não passa pelos pontos e define apenas uma tendência geral dos dados. Para um melhor resultado, recomenda-se a leitura das páginas de manual dos módulos.

Informações adicionais e mais detalhadas podem ser encontradas nos seguintes trabalhos: Smith & Wessel (1990); Mitasova & Mitas (1993); Mitasova & Hofierka (1993); Hofierka *et al.* (2002).

Módulos do GRASS para interpolação por RST:

- `v.surf.rst` – dados vetoriais (pontos, curvas de nível vetoriais)
- `v.vol.rst` – para geração de volumes tridimensionais (grid 3D)

10.4. Interpolação bilinear e bicúbica

Este método é bastante utilizado para a reamostragem de dados raster e para a geração de superfícies a partir de bandos de dados muito volumosos como os obtidos por altimetria laser:

- `r.resamp.interp` – reamostragem de dados raster
- `v.surf.bspline` – pontos vetoriais

10.5. Outros módulos de interpolação

Outros módulos do GRASS para interpolação de dados raster:

- `r.surf.contour` – Cálculo de superfícies (MDEs) a partir de curvas de nível em formato raster
- `r.surf.fractal` – Gera uma superfície fractal de determinada dimensão fractal
- `r.surf.gauss` – Usa um gerador de números aleatórios gaussiano para gerar uma superfície de valores de distribuição gaussiana com média e desvio padrão determinados pelo usuário
- `r.surf.random` – Gera um mapa raster com valores aleatórios com intervalo (*range*) definido pelo usuário.

10.6. Reamostragem dos dados SRTM

Vamos aproveitar a oportunidade e corrigir alguns problemas do mapa de aspecto que fizemos anteriormente. Como utilizamos o modelo SRTM para calcular o aspecto, e os dados originais estão em formato de números inteiros, temos artefatos nas direções 0° , 45° , 90° , 180° , 225° , 270° e 315° . Vamos fazer uma reamostragem dos dados originais mas sem alterar muito a resolução espacial. Isso já será suficiente para obter uma superfície um pouco mais contínua e com dados tipo *float*.

A mudança da resolução espacial para 25m já dever ser suficiente. Ainda no terminal, rode `g.region -pa region=petar res=25`. Agora abra o comando `r.resamp.interp` em **Raster** → **Develop raster map** → **Resample using multiple methods** [`r.resamp.interp`]. Nossa mapa de entrada será `srtm_30m` e o de saída `srtm_petar_25m_bilinear`. Aqui **bilinear** corresponde ao método de interpolação que pode ser escolhido na aba **Method**.

Com a reamostragem completa, refaça o mapa de aspecto, usando o mapa reamostrado como entrada e `srtm_petar_25m_bilinear_aspect` como saída. Examine o histograma de cada um desses mapas com a ferramenta **Analyze** → **Create histogram of image or raster file** do Map Display. Na figura 10.4 você vê os histogramas dos mapas de aspecto a partir dos dados originais e reamostrados.

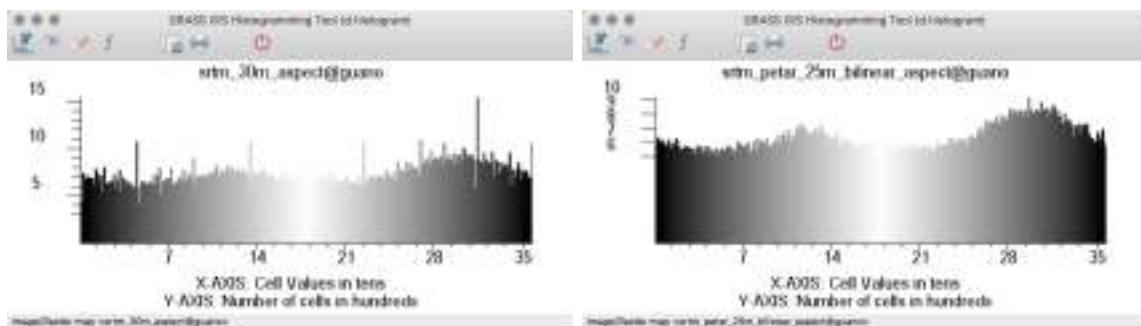


Figura 10.4. Histogramas dos mapas de aspecto, a partir dos dados SRTM originais (esquerda) e reamostrados com `r.resamp.interp` (direita).

10.7. Extração de curvas de nível

Agora que já temos o modelo SRTM corrigido, podemos extrair as curvas de nível do MDE. Esta é uma operação muito útil quando se trabalha em áreas carentes de levantamentos topográficos de maior detalhe. Usaremos o comando `r.contour`, disponível no menu **Raster** → **Terrain Analysis** → **Generate contour lines** [`r.contour`]. Como opções deste comando, podemos selecionar exatamente quais curvas de nível queremos ou indicar os níveis mínimo e máximo, e o intervalo altimétrico entre as curvas.

No exemplo usamos o raster `srtm_petar_25m_bilinear` e criamos o mapa vetorial `srtm_bilinear_contours`. Com base nas informações obtidas com o comando `r.info`, definimos como valor mínimo a cota de 0 m, e como máximo 1500 m, com intervalo entre as curvas de 20 m, o mesmo utilizado nos mapas topográficos em escala 1:50.000. Note que não é necessário usar valores que estejam dentro do intervalo de valores do mapa. No nosso caso, o valor mínimo do raster é de -50 m, mas utilizamos como cota mínima 0 m, para obter um espaçamento mais usual das curvas de nível.

Dentre os shapefiles que baixamos do SIG-Ribeira, importe o arquivo `Isolinhas` e o compare as curvas que foram geradas por `r.contour`. A figura 10.5 mostra os resultados obtido.

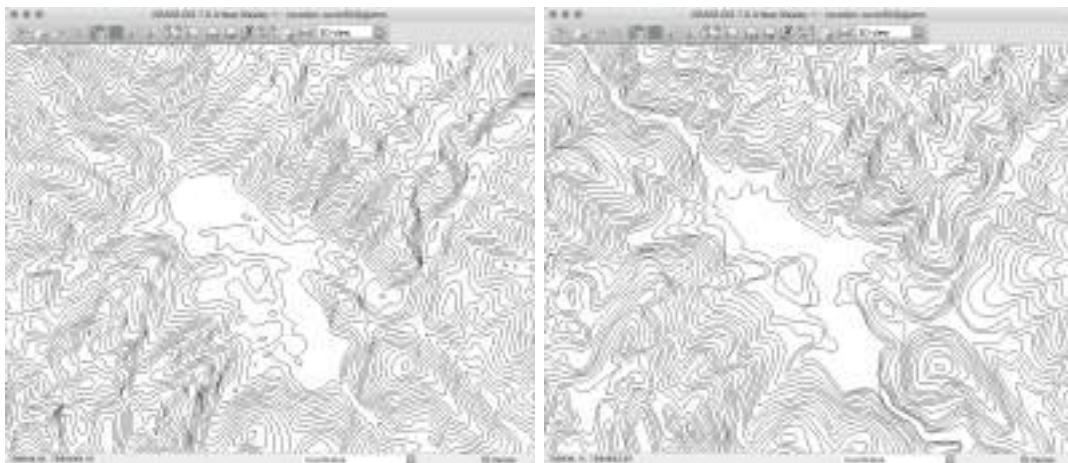


Figura 10.5. Detalhe das curvas de nível extraídas do SRTM para cotas definidas entre 0m e 1500m (esquerda) e comparação com curvas de escala 1:50.000 do SIG-Ribeira (direita).

10.8. Interpolação de MDE

10.8.1 Recorte de dados vetoriais

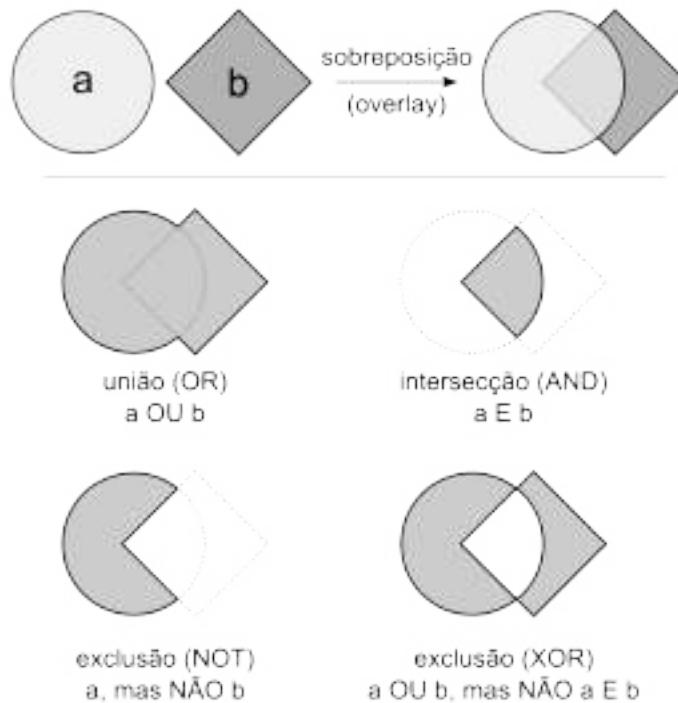
Antes de partir para a interpolação do DEM, vamos aproveitar e ver outra ferramenta importante, que usa lógica *booleana* para sobreposição de vetores. O comando `v.overlay` (**Vector → Overlay vector maps → Overlay vector maps [v.overlay]**) usa dois mapas vetoriais de entrada para produzir um mapa de saída a partir das operações (fig. 10.6):

and – operação de **intersecção**. Os elementos do mapa A e de mapa B.

or – operação de **união**. Os elementos de A **ou** de B.

not – operação de **exclusão**. Os elementos de A mas **não** de B.

xor – operação de **disjunção exclusiva** (eXclusive OR). Os elementos de A **ou** de B, **mas não** os de A e B

Figura 10.6. Operações de *overlay*.

Vamos usar a operação de **intersecção** para recortar as curvas de nível apenas na **region** que salvamos anteriormente (`g.region -p region=petar`).

Para o recorte precisamos de um polígono com os limites da **region**, e o comando `v.in.region` faz justamente isso. Vá em **Vector** → **Generate area for current region** [`v.in.region`] e crie o mapa **region_petar**.

Agora use `v.overlay`. Na aba **Required**, entre com o mapa `srtm_bilinear_contours` como entrada A (`ainput`), `region_petar` como entrada B (`binput`), `curvas_srtm_petar` como `output` e a operação de `overlay` como `and`. Na aba **Optional** defina o tipo de `ainput` como `line`. Essa operação pode demorar um pouco, já que o arquivo de entrada é bem grande.

Caso não tenha feito a importação dos outros vetores (`Pontos_Cotados` e `Drenagem`) considerando apenas os dados dentro da `region petar`, faça a operação de *overlay* para eles também.

O modelo de dados vetorial do GRASS permite, ao contrário dos *shapefiles*, mais de um tipo de geometria em um mesmo arquivo, como linhas e pontos (você já está vendo onde isso vai levar, certo?). Isso vai nos permitir fazer a interpolação usando as curvas de nível e os pontos cotados ao mesmo tempo, o que melhora bastante a qualidade do resultado final.

Para isso, é preciso juntar os dados que vieram em *shapefiles* separados. Assim como com os mapas *raster*, o GRASS chama essa operação de *patch*.

Mas as coisas nunca são tão simples como gostaríamos. Vamos ver as tabelas de atributos do mapas `Pontos_Cotados` e `curvas_srtm_petar`. No arquivo de pontos cotados, as cotas estão em uma coluna (a única) chamada `AVG_Z`, enquanto que no arquivo de curvas de nível temos várias colunas e a cota está na coluna `a_AVG_Z` (no arquivo original de isolinhas a coluna é `AVG_Z`, mas o nome foi alterado na operação de *overlay*). Uma solução simples é converter os dois mapas para 3D, onde o valor da cota é atribuído à geometria de cada elemento, independente da tabela de atributos.

O comando para fazer essa conversão é o `v.to.3d` (**File → Map type conversions → 2D vector to 3D vector [v.to.3d]**). Na aba **Required**, selecione o arquivo `Pontos_Cotados` como **input** e dê um nome ao arquivo de saída (`Pontos_Cotados_3d`, por exemplo). Na aba **Height**, selecione a coluna que contém os valores de elevação. Faça o mesmo para o mapa de isolinhas. Para checar se está tudo certo, visualize os metadados e procure pela informação: `Map is 3D: Yes`.

```
v.info map=curvas_pontos_petar_3d@guano
+-----+
| Name:          curvas_pontos_petar_3d
| Mapset:        guano
| Location:     cursoRC
| Database:     /Volumes/MacintoshHD2/grassdata
| Title:         Output from v.patch
| Map scale:    1:1
| Name of creator: guano
| Organization:
| Source date:   Thu May 19 22:44:59 2016
| Timestamp (first layer): none
|-----|
| Map format:   native
|-----|
| Type of map: vector (level: 2)
|
| Number of points:      333          Number of centroids:  0
| Number of lines:        2986         Number of boundaries: 0
| Number of areas:        0            Number of islands:    0
| Number of faces:        0            Number of kernels:    0
| Number of volumes:      0            Number of holes:      0
|
| Map is 3D:             Yes
| Number of dblinks:     0
|
| Projection: UTM (zone 22)
|
|           N: 7298380.3047343  S:          7271575
|           E: 751544.23063449  W: 720117.01911024
|           B:          59      T:          1160
|
| Digitization threshold: 0
| Comment:
|-----+
```

Com os dois mapas convertidos para 3D, podemos fazer a operação de *overlay*. O comando é **v.patch**, vá em **Vector** → **Overlay vector maps** → **Patch vector maps** [**v.patch**]. Defina os dois mapas que acabamos de converter como **input** e dê um nome descritivo ao arquivo de saída (como **curvas_pontos_petar_3d**).

Antes de iniciar a interpolação é sempre bom checar os limites da **region** de interesse, já que alguns parâmetros do comando têm seus valores padrão determinados a partir da resolução espacial da **region** ativa. Agora sim inicie o comando **v.surf.rst**, pelo menu **Raster** → **Interpolate surfaces** → **Regularized spline tension**. Este comando tem *muitas* opções.

Na janela do comando **v.surf.rst**, entre com **curvas_pontos_petar_3d** na aba **Required**. Na aba **Selection** escolhemos de onde buscar os valores de elevação: se os dados estiverem na tabela de atributos, deixe layer em 1 e na aba **Parameters** entre com o nome da coluna em **zcolumn**; se o mapa for 3D, pode deixar o layer em 1 e **zcolumn** em branco. A opção **WHERE** é utilizada para selecionar uma parte dos dados, como por exemplo: **income < 1000 and inhab > 10000**. Na aba **Output**, chame o mapa resultante de **petar_topografia_rst**, e mantenha os demais valores como os padrões sugeridos pelo programa. Ainda pode-se calcular ao mesmo tempo os mapas de declividade, orientação de vertentes e curvaturas (seção 9.3). Exiba o resultado no *Map Display* e compare com o modelo SRTM reamostrado por interpolação bilinear (fig. 10.7).

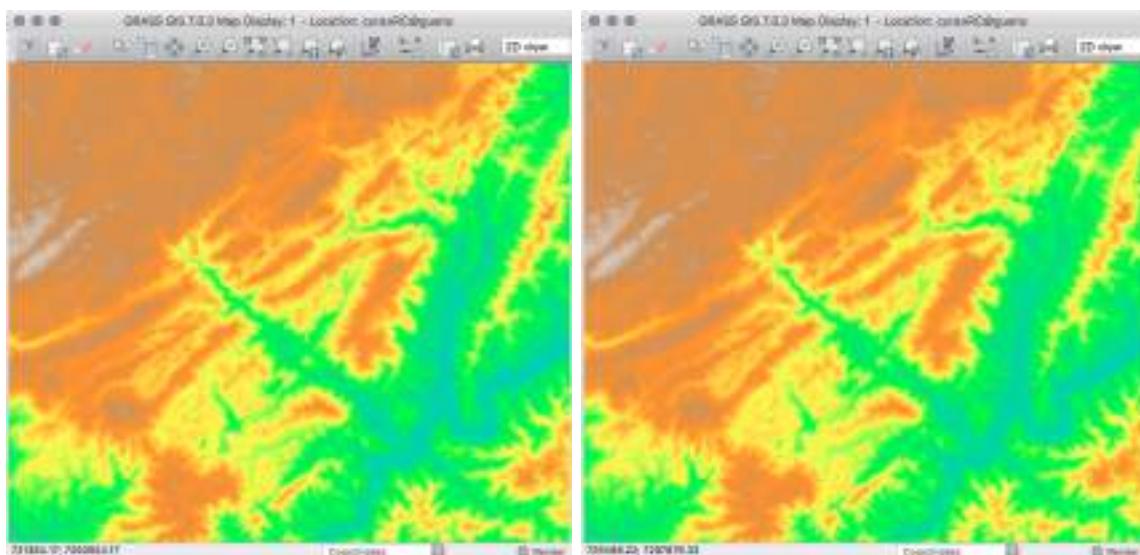


Figura 10.7. Resultado da interpolação por RST (esquerda) e modelo SRTM (direita).

Aparentemente os mapas são muito similares. As cores são diferentes porque no mapa interpolado, o intervalo de valores para distribuir as cores é menor. Vamos ver isso com mais detalhe usando a ferramenta **Map Calculator** para álgebra de mapas raster (comando `r.mapcalc`).

No mapcalc, faça uma subtração dos dois mapas (SRTM menos RST) e chame o resultado de `srtm_diff_rst`. Use `r.colors` para aplicar a escala de cores `difference` e exiba o resultado (fig. 10.8).

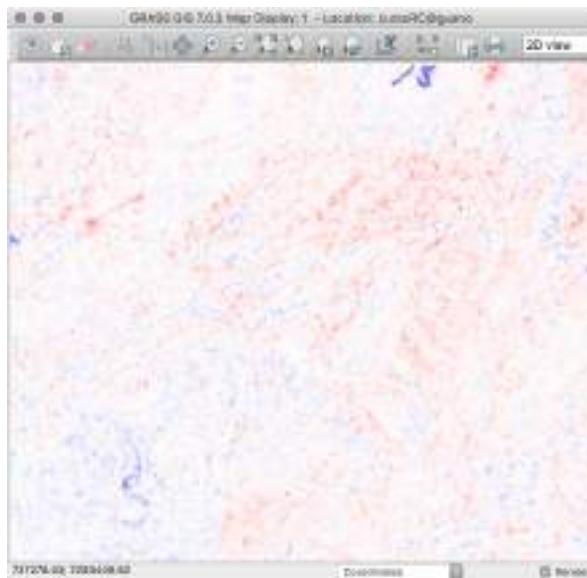


Figura 10.8. Resultado da subtração entre o modelo SRTM e a interpolação por RST.

Vamos ver as diferenças numéricas com estatísticas para cada mapa com o comando `r.univar`:

<code>r.univar map=srtm_petar_25m_bilinear@guano</code>	<code>r.univar map=petar_topografia_rst@guano</code>
total null and non-null cells: 1301870	total null and non-null cells: 1301870
total null cells: 0	total null cells: 0
Of the non-null cells:	Of the non-null cells:
-----	-----
<code>n: 1301870</code>	<code>n: 1301870</code>
<code>minimum: 27.9097</code>	<code>minimum: 39.4679</code>
<code>maximum: 1190.17</code>	<code>maximum: 1168.05</code>
<code>range: 1162.26</code>	<code>range: 1128.58</code>
<code>mean: 577.077</code>	<code>mean: 570.991</code>
<code>mean of absolute values: 577.077</code>	<code>mean of absolute values: 570.991</code>
<code>standard deviation: 270.744</code>	<code>standard deviation: 271.833</code>
<code>variance: 73302.3</code>	<code>variance: 73893.3</code>
<code>variation coefficient: 46.9165 %</code>	<code>variation coefficient: 47.6072 %</code>
<code>sum: 751278788.243046</code>	<code>sum: 743356399.74107</code>

```
r.univar map=srtm_diff_rst@guano
total null and non-null cells: 1301870
total null cells: 0
Of the non-null cells:
-----
n: 1301870
minimum: -183.262
maximum: 216.266
range: 399.528
mean: 6.08539
mean of absolute values: 13.8398
standard deviation: 17.4798
variance: 305.542
variation coefficient: 287.241 %
sum: 7922388.50198572
```

Você também pode visualizar as diferenças com o *Map Display*. Na janela das propriedades do *raster* (que é o comando `d.rast`), na aba **Selection**, você pode entrar com valores no formato **mínimo-máximo**, como **-30-30** (de -30 até +30) ou **-30- -15** (de -30 até -15) e ver o resultado no *Map Display*. Para mostrar todos os valores maiores que 50, por exemplo, não podemos colocar **>50**. É preciso um intervalo de valores, então podemos colocar um limite superior maior que o máximo, como em **50-250** (Fig. 10.9).

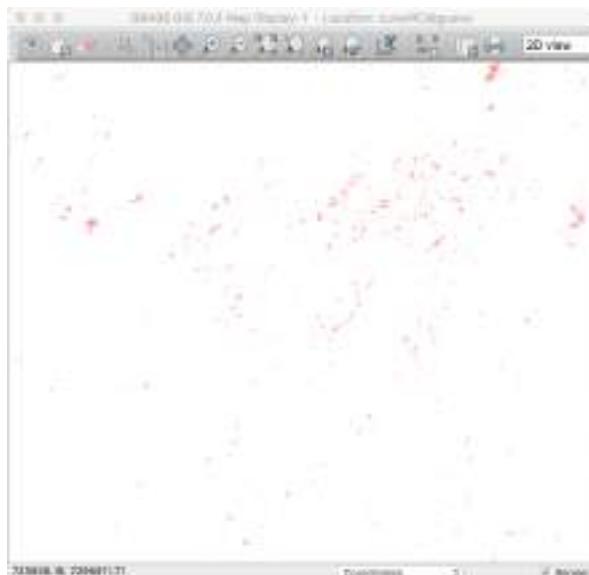


Figura 10.9. Diferenças entre os mapas, exibindo valores maiores que 50m.

Outra maneira de visualizar a diferença entre os mapas é exibir um histograma (Fig. 10.10). No *Map Display*, procure o ícone de **Analize map** (logo após os ícones de zoom) e escolha uma das opções de histograma. Por padrão, o mapa que estiver selecionado no *Layer Manager* será analisado. A primeira opção, **Create histogram of raster map**, usa a biblioteca Matplotlib²⁴ para plotar o gráfico, e oferece um resultado mais bonito e com várias opções de configuração (tamanho da fonte etc) e permite salvar o arquivo em formatos vetoriais de ilustração (como .pdf e .svg). A opção **Create histogram with d.histogram** usa uma saída gráfica mais simples e não permite salvar o gráfico em formato vetorial, apenas como uma imagem .png. Note que as cores do histograma são as mesmas do mapa raster.

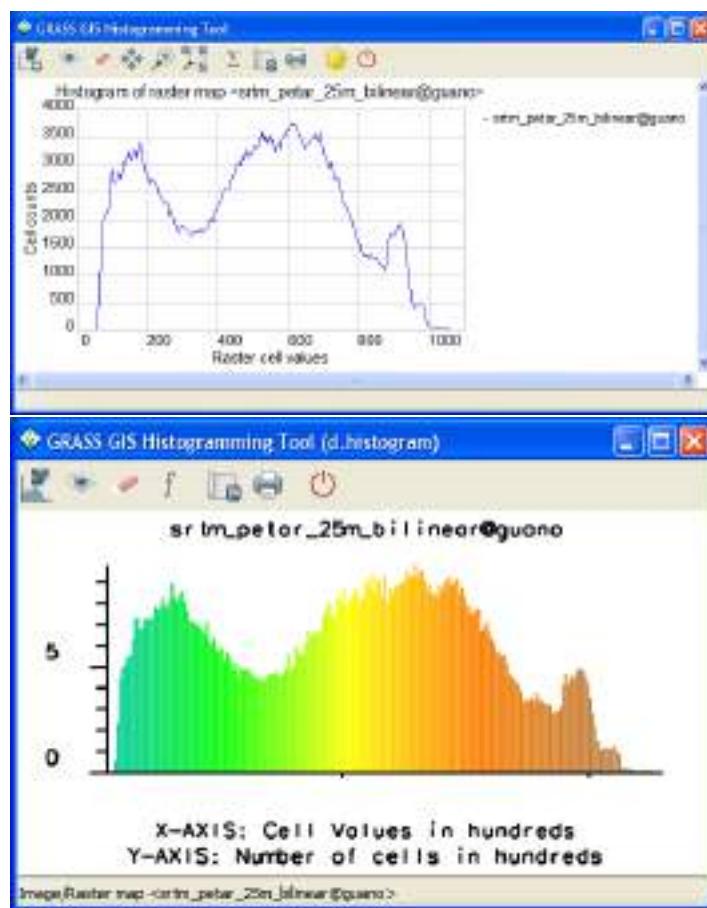


Figura 10.10. Histograma do mapa de diferenças. Acima: com uso da biblioteca Matplotlib. Abaixo: saída do comando `d.histogram`.

²⁴<http://matplotlib.org>

11. Processamento Digital de Imagens

11.1. Aquisição das imagens

Neste capítulo veremos um pouco de Processamento Digital de Imagens (PDI). Vamos usar como exemplo uma imagem Landsat 8 OLI.

Vamos começar com o download dos dados. Existem vários sites que podemos usar, como o EarthExplorer²⁵, Glovis²⁶ ou o catálogo do INPE²⁷, mas aqui vamos fazer algo um pouco diferente.

O catálogo do Landsat 8 está disponível no *Amazon Web Services*²⁸ (AWS), então usuários desse sistema de computação em nuvem podem processar os dados utilizando esse serviço. Mas também podemos acessar as imagens diretamente.

Os dados estão organizados no AWS em uma estrutura de diretórios de acordo com WRS-2 de órbita/ponto (*path/row*). Por exemplo, a cena LC81390452014295LGN00 está no endereço: s3://landsat-pds/L8/139/045/LC81390452014295LGN00/. O diretório L8 significa Landsat 8, 139 é a órbita (*path*) e 045 o ponto (*row*). A parte final do nome do diretório é o identificador da cena na convenção LXSPPPRRYYYYDDDGSIIVV, onde:

```

L = Landsat
X = Sensor
S = Satellite
PPP = WRS path
RRR = WRS row
YYYY = Year
DDD = Julian day of year
GSI = Ground station identifier
VV = Archive version number
  
```

Então nosso primeiro passo é descobrir qual cena (órbita/ponto) recobre nossa área de interesse. Na página https://landsat.usgs.gov/tools_wrs-2_shapefile.php encontramos um arquivo *shapefile* com os polígonos do WRS-2. Baixe o arquivo *wrs2_descending.zip*, descompacte-o e abra-o no QGIS. Abra também o arquivo *S25W049.hgt*. Veja que a cena 220/77 recobre muito bem a área que estamos trabalhando.

Depois de descobrir qual órbita/ponto precisamos, vamos ver quais cenas estão disponíveis, e se elas têm a qualidade que precisamos (como baixa cobertura de nuvens, época do ano etc.). O arquivo http://landsat-pds.s3.amazonaws.com/scene_list.gz traz uma listagem de todas as cenas armazenadas no AWS. É um arquivo CSV (*comma separated value*) de texto simples com campos separados por vírgulas, que pode ser aberto pelo Excel ou OpenOffice Calc, por exemplo.

²⁵<http://earthexplorer.usgs.gov>

²⁶<http://glovis.usgs.gov>

²⁷<http://www.dpi.inpe.br/cdteca/>

²⁸https://aws.amazon.com/?nc2=h_lg

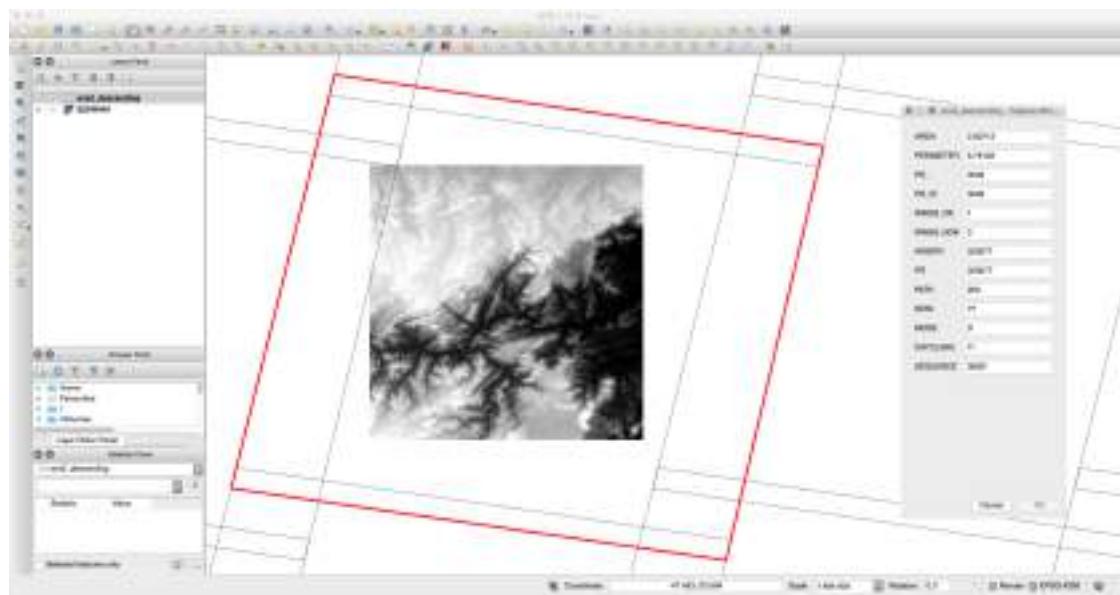


Figura 11.1. Determine a cena WRS-2 que recobre a área de estudo.

Baixe o arquivo `scene_list.gz` (cerca de 21,5 MB) e descompacte-o. O arquivo descompactado não tem extensão, então programas como o Excel podem não reconhecê-lo. Simplesmente renomeie-o e adicione uma extensão `.csv` ou `.txt` no final. No Excel, vá em **Data** → **Get external data** → **Import text file** e selecione-o. No diálogo de importação dos dados, escolha texto **Delimitado** e na sequência, marque o delimitador como **vírgula** (fig. 11.2).

A listagem é bem grandinha (quase 500.000 linhas!) e deve demorar um pouco para abrir. Uma vez aberto, ordene a planilha (**Data** → **Sort**) pelas colunas de `path`, depois `row`, depois `cloudCover`. Role a tela até encontrar a órbita 220 e ponto 77 (fig. 11.3). As cenas disponíveis estão ordenadas de acordo com a cobertura de nuvens (terceira coluna). O último campo da planilha é um *link* direto para os dados. Copie esse *link* da primeira cena da nossa lista, que tem apenas 2% de cobertura de nuvens, e cole-o em um navegador de Internet.

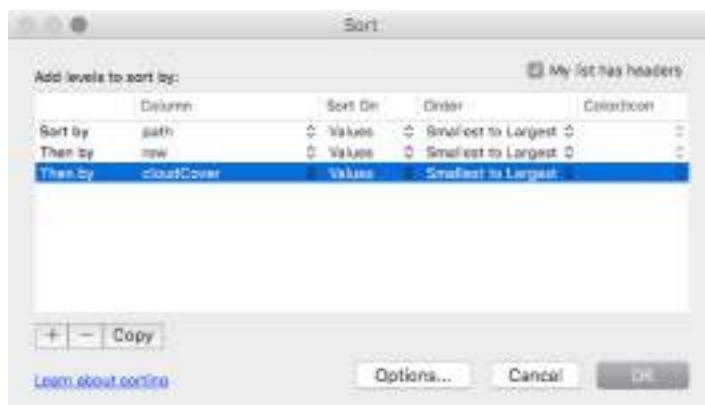


Figura 11.2. Ordenação da planilha com as cenas disponíveis.

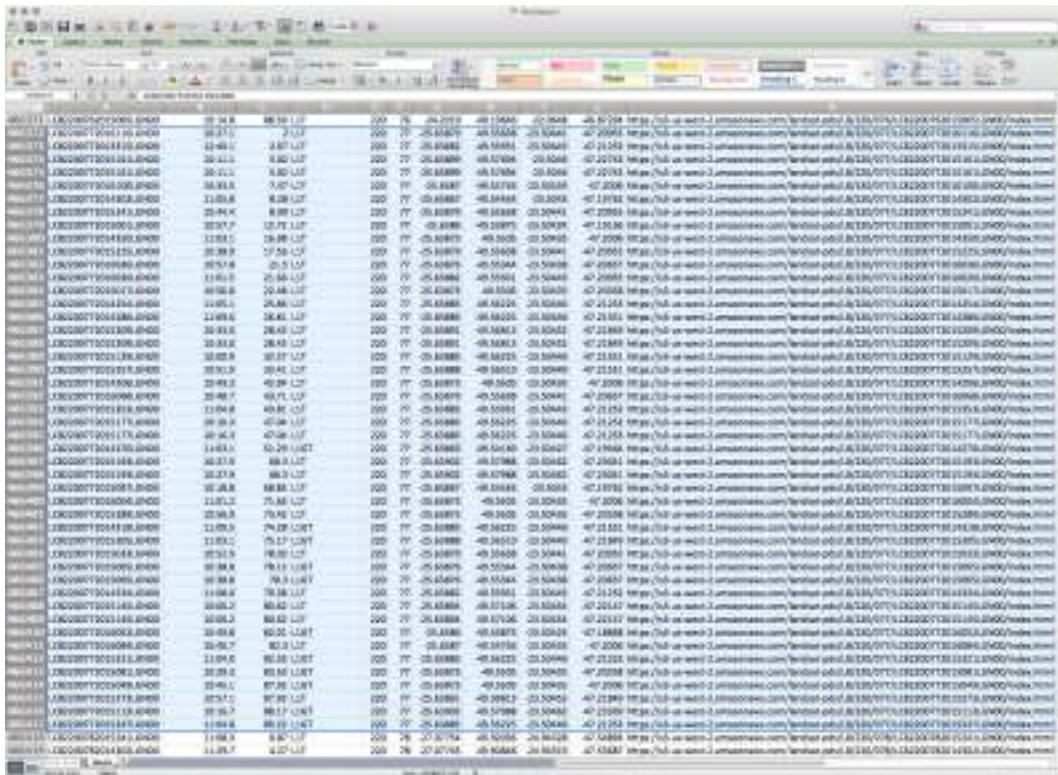


Figura 11.3. Cenas disponíveis, ordenadas de acordo com a cobertura de nuvens.

No navegador vemos os arquivos disponíveis para essa cena. A imagem na parte superior é um preview, e é uma maneira rápida de avaliar se a cena é de interesse ou não (por vezes, há uma cobertura de nuvens razoável mas a área de estudo é pequena e está em uma porção sem nuvens). No caso dessa primeira tentativa, infelizmente nossa região de análise está coberta de nuvens, então tentamos a cena seguinte. A cena LC82200772013315LGN00 tem apenas algumas nuvens na porção oeste e será nossa escolhida (fig. 11.4). Podemos baixar os arquivos com extensão .TIF e o arquivo que termina em _MTL.txt (os metadados dessa imagem).

Na hora de for escolher as cenas, também deve-se levar em conta o nível de processamento de cada uma. A página http://landsat.usgs.gov//Landsat_Processing_Details.php traz mais detalhes, mas de maneira simplificada cada imagem passa por um pré-processamento, que inclui correção da projeção cartográfica entre outros. Dê preferências às imagens com nível 1T de processamento, que têm correções geométricas (georreferenciamento) com apoio de modelos de elevação e de pontos de controle em solo.

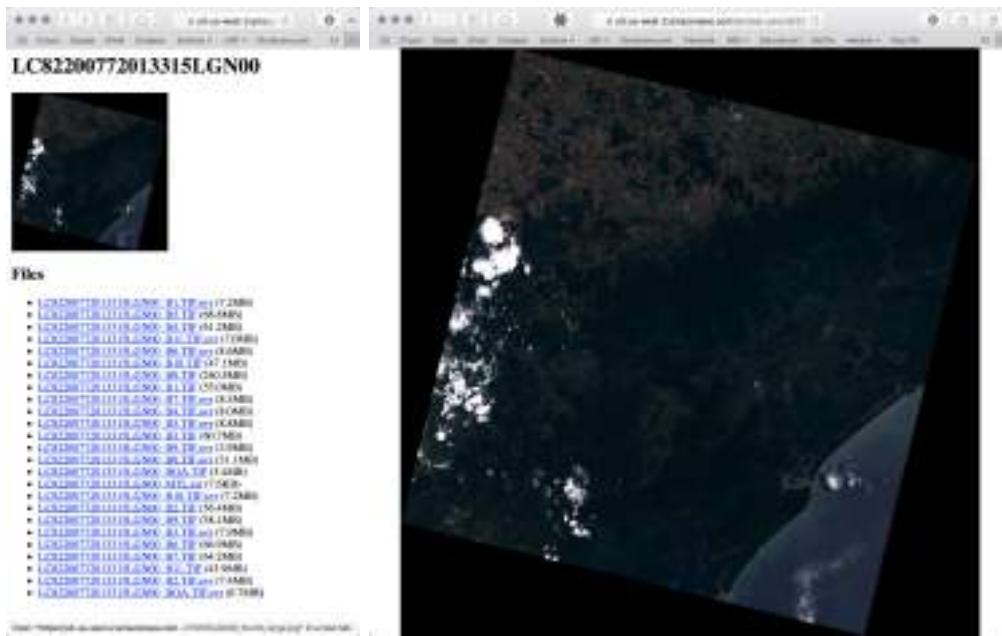


Figura 11.4. Arquivos da cena LC82200772013315LGN00 (esquerda) e *preview* (direita).

11.2. Importação das imagens no GRASS

Cada arquivo que baixamos corresponde a uma banda do sensor OLI (*Operational Land Imager*). As bandas representam faixas do espectro de radiação eletro-magnética (REM) e não iremos importar todas, apenas as que nos interessam. A figura 11.5 mostra as bandas do sensor OLI (Landsat 8) e ETM+ (Landsat 7)²⁹.

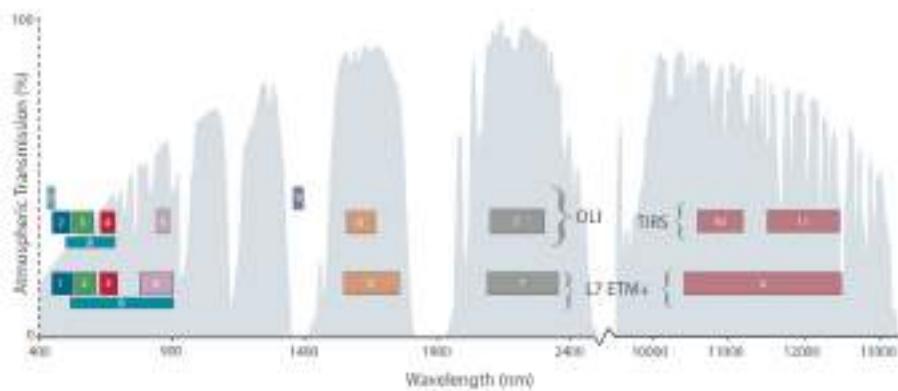


Figura 11.5. Bandas dos sensores ETM+ (Landsat 7) e OLI (Landsat 8).

²⁹http://landsat.gsfc.nasa.gov/wp-content/uploads/2013/01/ETM+vOLI-TIRS-web_Feb20131.jpg

Houve uma renomeação das bandas no sensor OLI (o que incomodou muita gente). A figura 11.6 traz uma tabela³⁰ com as características das bandas e seus comprimentos de onda.

Landsat-7 ETM+ Bands (μm)			Landsat-8 OLI and TIRS Bands (μm)		
Band 1	30 m Blue	0.441 - 0.514	30 m Coastal/Aerosol	0.435 - 0.451	Band 1
Band 2	30 m Green	0.519 - 0.601	30 m Green	0.533 - 0.590	Band 2
Band 3	30 m Red	0.631 - 0.692	30 m Red	0.636 - 0.673	Band 3
Band 4	30 m NIR	0.772 - 0.898	30 m NIR	0.851 - 0.879	Band 4
Band 5	30 m SWIR-1	1.547 - 1.749	30 m SWIR-1	1.566 - 1.651	Band 5
Band 6	60 m TIR	10.31 - 12.36	100 m TIR-1	10.60 - 11.19	Band 10
			100 m TIR-2	11.30 - 12.51	Band 11
Band 7	30 m SWIR-2	2.064 - 2.345	30 m SWIR-2	2.107 - 2.294	Band 7
Band 8	15 m Pan	0.515 - 0.896	15 m Pan	0.503 - 0.676	Band 8
			30 m Cirrus	1.363 - 1.384	Band 9

Figura 11.6. Bandas e comprimentos de onda dos sensores ETM+ (Landsat 7) e OLI (Landsat 8).

Abra uma das bandas no QGIS e cheque os metadados: a projeção deve estar como EPSG:32622, WGS84 / UTM Zone 22N. Mas se está em UTM para o hemisfério Norte, porquê a imagem é exibida na posição correta no QGIS? Porque sistema de produção de imagens do USGS atribui um valor negativo para as coordenadas Y de imagens no hemisfério Sul. Isso funciona no QGIS (e no ArcGIS), mas não no GRASS, que espera valores positivos para essas coordenadas.

Então precisamos fazer uma transformação simples, usando um utilitário de linha de comando da biblioteca GDAL. No terminal, navegue até o diretório onde as imagens estão salvas. Para cada banda, use o seguinte comando:

```
gdalwarp -s_srs EPSG:32622 -t_srs EPSG:32722 <input> <output>
```

Isso vai transformar as coordenadas do sistema *source* (-s_srs EPSG:32622) para o sistema *target* (-t_srs EPSG:32722). Para o arquivo de saída, repita o nome do arquivo de entrada e adicione alguma informação no final, como *_south*.

Agora que as imagens estão em coordenadas UTM Sul, vá em **File** → **Import raster data** → **Common import formats [r.in.gdal]**. Em **Source type** selecione **directory** e escolha a pasta onde os arquivos estão. O formato **GeoTIFF** já deve estar selecionado como padrão, e uma listagem dos arquivos com esse formato deve ser carregada automaticamente. Selecione as bandas para importar (mude o nome que o layer vai receber dentro do GRASS com um duplo-clique sobre o nome e retire o *_south*) e rode o comando. Importe todas as bandas.

³⁰http://landsat.gsfc.nasa.gov/wp-content/uploads/2013/01/BandpassesL7vL8_Jul20131.pdf

11.3. Pré-processamento

Vamos começar com uma composição colorida RGB. No Layer Manager, vá em **Add various raster maps** → **Add RGB map layer**. Para fazer uma composição “cor natural” usamos as bandas 4 em **red** , 3 em **green** e 2 em **blue**. Ficou bom? Provavelmente não. Vamos melhorar um pouco as cores. Vá em **Imagery** → **Manage image colors** → **color balance for RGB [i.colors.enhance]** Seleccione as mesmas bandas da composição colorida (4/3/2) e na aba **Optional** altere o valor do limiar para 99. Atualize o Map Display e veja o resultado (fig. 11.7).

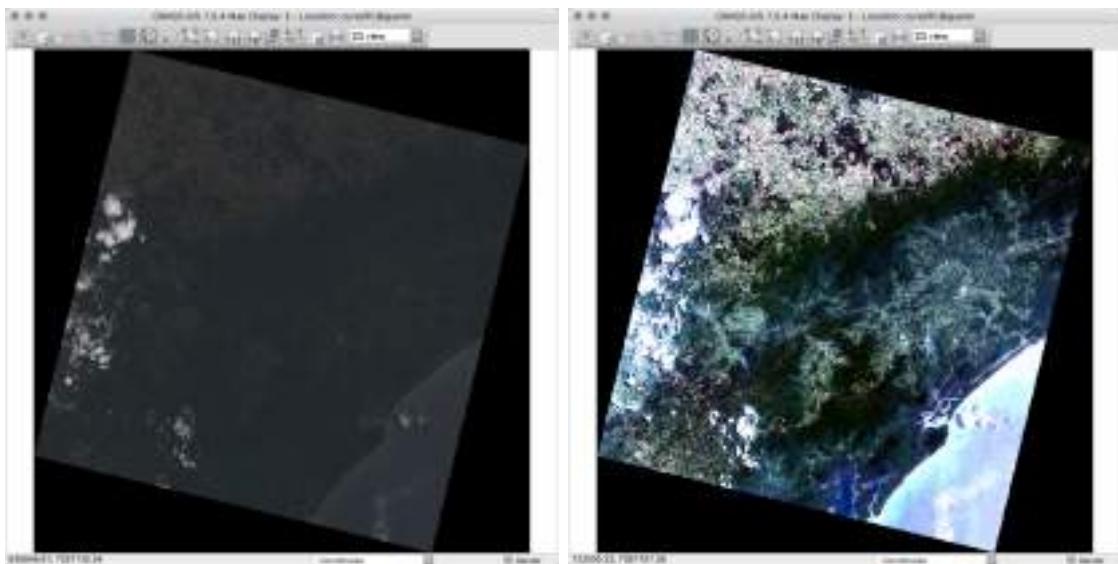


Figura 11.7. Composição colorida $\text{RGB}=4/3/2$, antes (esquerda) e depois (direita) do contraste das cores.

Mas o ideal é fazer uma etapa de pré-processamento das imagens que consiste em converter os valores dos pixels de DN (*digital number*, valores quantizados no sensor) para reflectância ou radiância. Isso é feito com o comando **i.landsat.toar** (**Imagery** → **Satellite images tools** → **Landsat DN to radiance/reflectance [i.landsat.toar]**).

A opção **input** é a parte do nome dos layers que é comum a todos (o comando vai processar todas as bandas da imagem): **LC82200772013315LGN00_B**, e **output** será o nome base que todos os layers vão receber. O comando vai adicionar os números das bandas após esse prefixo. Use **LC82200772013315LGN00_ref1**. (veja o ponto depois de **ref1**). Na aba **Metadata**, encontre o arquivo com final **_MTL.txt** para a opção **metfile**, selecione **oli8** em **sensor**, deixe o método de correção atmosférica como **uncorrected** e a data de aquisição da imagem pode ser encontrada no arquivo **_MTL.txt**. Procure por **DATE_ACQUIRED** (nesse caso, **2013-11-11**). Rode o comando.

Visualmente o resultado não é muito diferente do original, exceto que os mapas resultantes do **i.landsat.toar** não têm a borda preta ao redor (fig. 11.9). Veja os metadados de uma das bandas e compare as diferenças:

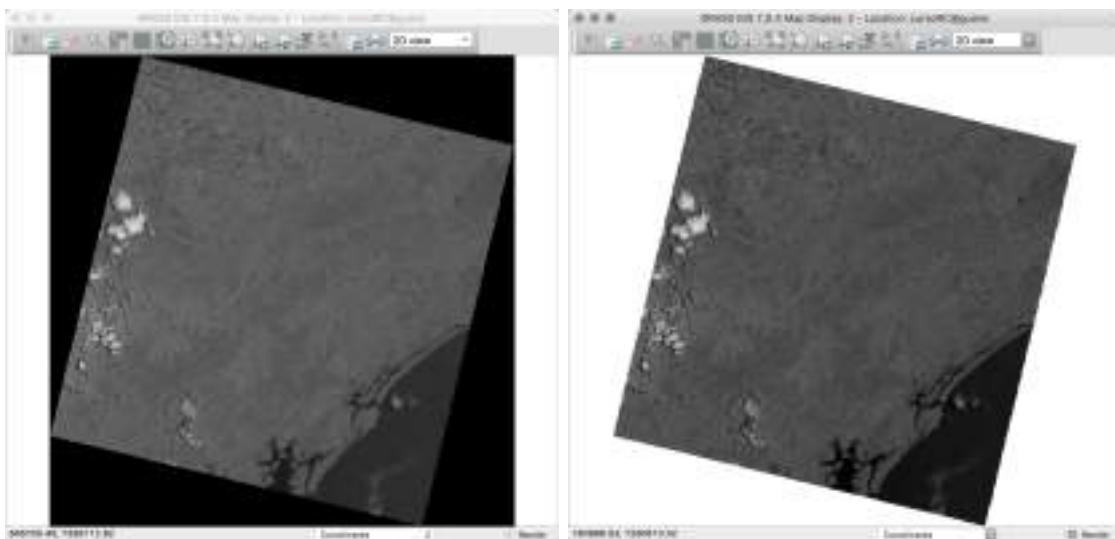


Figura 11.8. Banda 5 do OLI, antes (esquerda) e depois (direita) do `i.landsat.toar`.

```
+-----+
| Map:      LC82200772013315LGN00_B5@guan  Date: Mon May 23 12:53:53 2016 |
| Mapset:   guano                         Login of Creator: guano      |
| Location: cursoRC                      |
| DataBase: /Volumes/MacintoshHD2/grassdata |
| Title:    ( LC82200772013315LGN00_B5 )   |
| Timestamp: none                         |
+-----+
|
|
| Type of Map: raster          Number of Categories: 0
| Data Type:   CELL
| Rows:        7881
| Columns:    7781
| Total Cells: 61322061
| Projection: UTM (zone -22)
|           N: 7399915.00000511   S: 7163485.00000184   Res:    30
|           E: 880515.00000325   W: 647085.00000003   Res:    30
| Range of data:   min = 0   max = 65535
|
|
| Data Description:
|   generated by r.in.gdal
|
|
| Comments:
|   r.in.gdal --overwrite input="/Volumes/MacintoshHD2/Dropbox/USP/aulas\
|   /cursos_palestras/curso_GRASS_RC_Eco/dados/2015_01_14_OLI/LC82200772\
|   013315LGN00_B5_south.tif" output="LC82200772013315LGN00_B5" memory=3\
|   00 offset=0
|
+-----+
```

```
+-----+
| Map:      LC82200772013315LGN00_refl.5@ Date: Mon May 23 12:59:19 2016   |
| Mapset:   guano                      Login of Creator: guano          |
| Location: cursoRC                   |
| DataBase: /Volumes/MacintoshHD2/grassdata |
| Title:    ( LC82200772013315LGN00_refl.5 ) |
| Timestamp: none                     |
|-----|
|
| Type of Map: raster           Number of Categories: 0   |
| Data Type:   DCELL            |
| Data Units:  unitless         Vertical datum: (none)   |
| Rows:        7881              |
| Columns:    7781              |
| Total Cells: 61322061        |
| Projection: UTM (zone -22)   |
|      N: 7399915.00000511     S: 7163485.00000184   Res:    30   |
|      E: 880515.00000325     W: 647085.00000003   Res:    30   |
| Range of data:   min = -0.00872317925628258   max = 1.34024702999528 |
|
| Data Description:           |
| generated by i.landsat.toar |
|
| Comments:                  |
| Reflectance of Landsat-8 OLI/TIRS (method uncorrected) |
|-----|
| Acquisition date (and time) ..... 2013-11-11 (13.2111 h) |
| Production date ..... 2015-01-14 |
|
| Earth-sun distance (d) ..... 0.9899637 |
| Sun elevation (and azimuth) ..... 64.60075 (79.41872) |
| Digital number (DN) range ..... 1 to 65535 |
| Calibration constants (Lmin to Lmax) ... -31.18599 to +377.64429 |
| DN to Radiance (gain and bias) ..... +0.00624 and -31.19223 |
| Mean solar irradiance (ESUN) ..... 960.362 |
| Radiance to Reflectance (divide by) ... +281.77215 |
|-----|
|
| i.landsat.toar --overwrite input="LC82200772013315LGN00_B" output="LC\ |
| C82200772013315LGN00_refl." metfile="/Volumes/MacintoshHD2/Dropbox/U\ |
| SPaulas/cursos_palestras/curso_GRASS_RC_Eco/dados/2015_01_14_OLI/LC\ |
| 82200772013315LGN00_MTL.txt" sensor="oli8" method="uncorrected" date\ |
| ="2013-11-11" percent=0.01 pixel=1000 rayleigh=0.0 scale=1.0 |
|
+-----+
```

Como a imagem é maior do que nossa área de interesse, vamos cortá-la, usando o `r.mapcalc`. selecione uma área um pouco maior do que a `region` que salvamos anteriormente e defina uma nova `region` a partir do `display`. vamos fazer esse recorte com a expressão `if(x, a, b)`, que já usamos antes. A novidade aqui é que vamos apenas testar a existência dos dados dentro da `region`, e todo o resto será definido como nulo, `null()`. Fica mais rápido fazer no terminal:

```
r.mapcalc "B1_clip = if(LC82200772013315LGN00_refl.1, LC82200772013315LGN00_refl.1, null())"
```

11.4. Índices de vegetação

Com as imagens de reflectância calculadas, podemos fazer mapas de índices de vegetação. O comando `i.vi` (**Imagery** → **Satellite images productos** → **Vegetation indicies** [`i.vi`]) calcula diversos tipos desses índices, a partir das imagens de reflectância. Como dados de entrada, temos que definir as bandas da nossa imagem:

<code>red</code>	banda 4
<code>nir</code>	banda 5
<code>green</code>	banda 3
<code>blue</code>	banda 2
<code>5th channel</code>	banda 6
<code>7th channel</code>	banda 7

A figura abaixo traz dois exemplos, um NDVI e um EVI. As escalas de cores foram inseridas na opção **Add map elements** do *Map Display*.

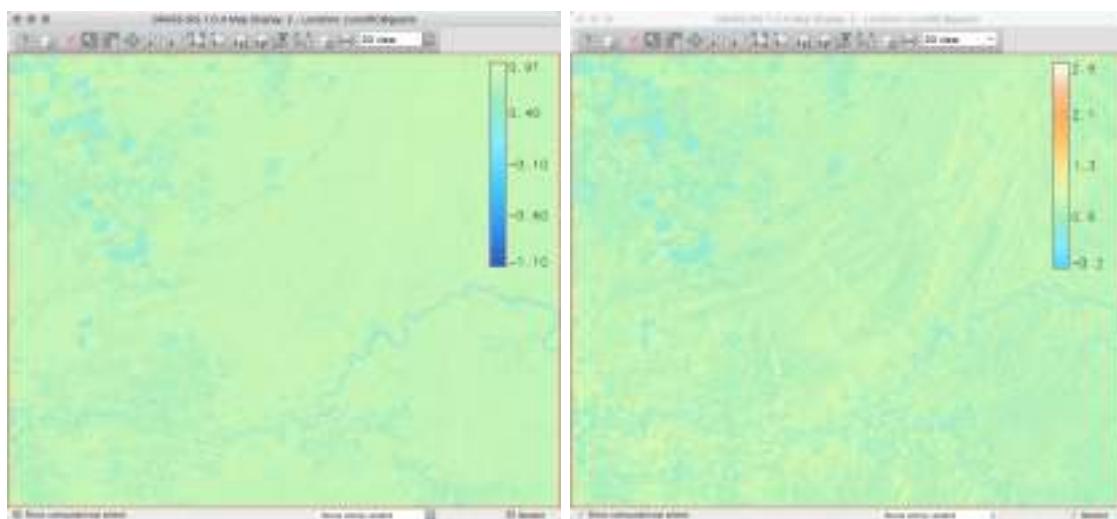


Figura 11.9. NDVI (esquerda) e EVI (direita).

11.5. Classificação

A classificação no GRASS é realizada em dois passos. Em uma classificação não-supervisionada roda-se primeiro o comando `i.cluster`, que vai gerar um arquivo com médias e matrizes de covariância para os pixels de uma imagem. Em uma classificação supervisionada, o comando a ser utilizado é o `g.gui.iclass`, onde o usuário define, interativamente, áreas que serão utilizadas para gerar assinaturas espectrais. Em ambos os casos, o segundo passo é o comando `i.maxlik`, que classifica as imagens por máxima verossimilhança baseado nos arquivos de assinatura gerados nos passos anteriores.

Vamos ver o processo com uma classificação não-supervisionada da nossa área de trabalho. Primeiro criamos um grupo e um subgrupo de imagens apenas com as bandas do visível e infra-vermelho (veja que não há espaços entre os nomes dos mapas e que a barra invertida significa que o comando continua na linha a seguir):

```
i.group group=lsat8 subgroup=lsat8 input=LC82200772013315LGN00_ref1.2,\  
LC82200772013315LGN00_ref1.3,LC82200772013315LGN00_ref1.4,\  
LC82200772013315LGN00_ref1.5,LC82200772013315LGN00_ref1.6,\  
LC82200772013315LGN00_ref1.8
```

A seguir rodamos `i.cluster` para gerar o arquivo de assinaturas espectrais:

```
i.cluster group=lsat8 subgroup=lsat8 signaturefile=sig_cluster_lsat8 \  
classes=10 reportfile=rep_clust_lsat8.txt
```

E em seguida, `i.maxlik` para a classificação:

```
i.maxlik group=lsat8 subgroup=lsat8 signaturefile=sig_cluster_lsat8 \  
output=lsat8_cluster_classes reject=lsat8_cluster_reject
```

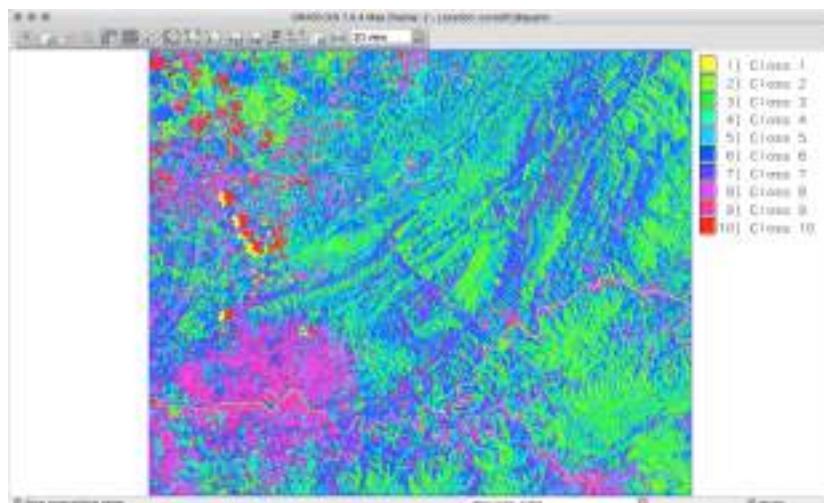


Figura 11.10. Classificação não-supervisionada, com 10 classes.

12. Visualização n-dimensional

Nesta seção veremos um pouco sobre o visualizador n-dimensional do GRASS, o NVIZ. Nele é possível visualizar superfícies (pixels - 2.5D) e volumes (voxels - 3D), bem como dados vetoriais.

A experiência mostra que é melhor trabalhar no NVIZ em **Locations** definidas com sistemas de coordenadas planas, pois não há necessidade de conversão dos valores de altitude. O NVIZ usa as definições da **region** ativa, e pode demorar para processar os dados dependendo do tamanho da área e resolução espacial. Escolha uma sub-área no mapa com a ferramenta **Zoom** e defina as definições da **region** a partir do *Map Display*.

Inicie o NVIZ alterando o modo de visualização do *Map Display* de 2D para 3D, e os mapas que estiverem com a visualização habilitada no *Layer Manager*. Uma nova aba irá aparecer no *Layer Manager*, com diversas opções para alterar a visualização 3D.

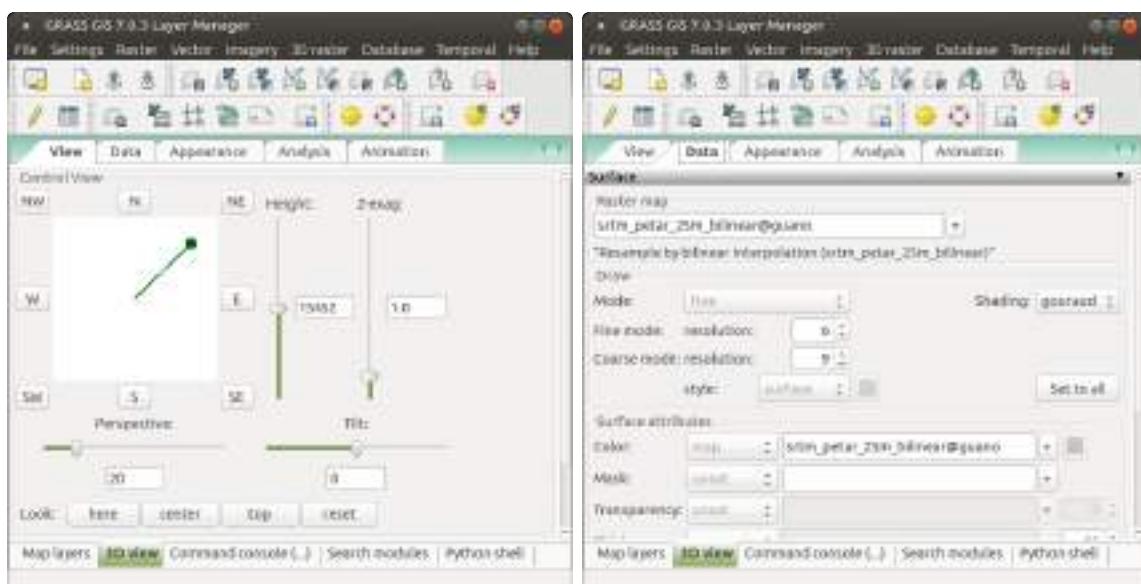


Figura 12.1. Janelas do *Layer Manager* com as opções de visualização 3D.

A posição do “observador” pode ser controlada arrastando a seta para que ela aponte para a direção que se deseja observar. A perspectiva aproxima ou afasta a cena, a altura (*height*) modifica sua posição na vertical e a opção *Z-exag* altera a escala vertical da superfície. Experimente um pouco os comandos para se acostumar com eles. Lembre-se que quanto maior a resolução espacial da **region** ativa, mais “pesado” fica para o programa.

Na aba **Appearance**, existem várias opções de iluminação, como cor e intensidade. Na aba **Data**, é possível inserir/modificar mapas (raster, vetoriais) bem como utilizar um raster como “cor” da superfície.

Vamos alterar a cor da superfície exibida. Na aba **Data**, em **Surface attributes**, o item **Color** pode ser uma cor única (**constant**) ou pode ser um *raster* (**map**).

Selecione o *raster* de diferenças entre o SRTM e o mapa interpolado por RST (**srtm_diff_rst**). Utilize os comandos da aba **View** para ajustar a visualização a seu gosto.

Na figura 12.2 vemos o resultado da sobreposição do mapa de diferenças entre o modelo SRTM e o interpolado por RST sobre a topografia, onde pode-se ver bem em quais áreas estão as maiores diferenças entre as superfícies. Veja que qualquer mapa raster pode ser sobreposto a outro no NVIZ, seja um modelo de terreno, imagem de satélite ou fotografia aérea.

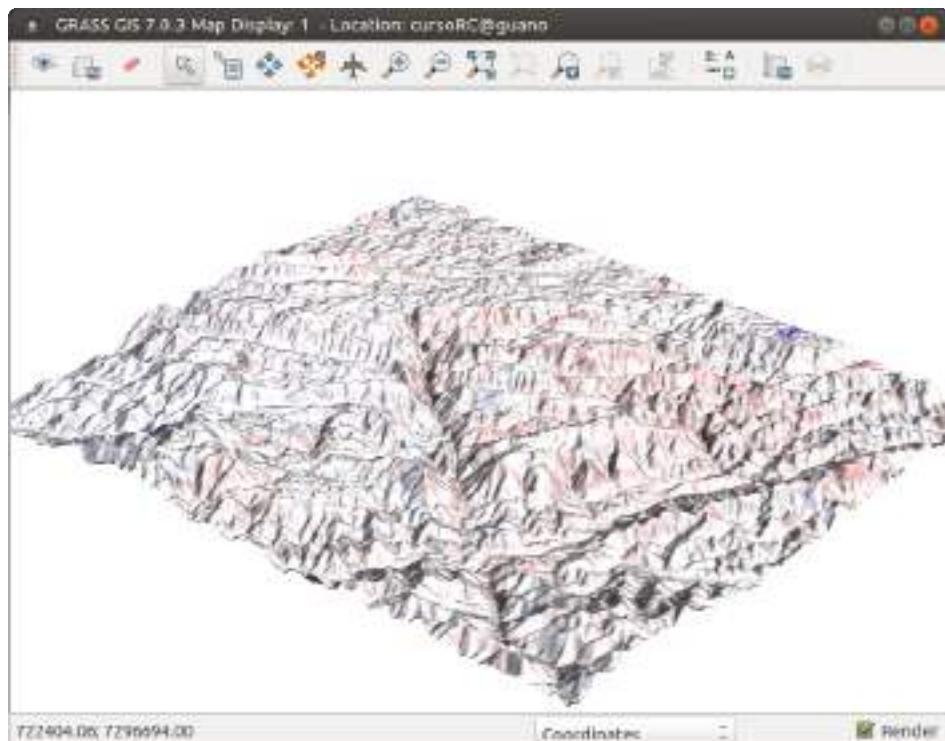


Figura 12.2. Sobreposição do mapa de diferenças à superfície topográfica.

13. Produzindo mapas para impressão

O comando `ps.map` permite que criemos mapas complexos, que serão salvos em arquivos tipo *postscript*³¹. As informações necessárias para construir o mapa (escala, grade de coordenadas, etc) são salvas em um arquivo de texto simples, o que permite que criemos vários mapas com exatamente o mesmo layout, apenas alterando o(s) layer(s) raster ou vetoriais desejados.

Nas versões mais atuais do GRASS, o comando `g.gui.psmap` **File → Cartographic composer [g.gui.psmap]** propicia ao usuário uma interface gráfica para disposição dos itens (mapa, escala, legenda, etc) na página do mapa, depois gera um arquivo de instruções e roda o `ps.map` em segundo plano.

Abaixo vemos um exemplo de um script simples para `ps.map`. O comando possui diversas opções, que podem ser consultadas na sua página de manual. Salve o texto abaixo em um arquivo texto e rode o comando `ps.map` no terminal para abrir sua janela.

```
rast srtm_petar_25m_bilinear

vlines Drenagem
color blue
width 0.1
end

grid 5000
color black
numbers 2
end

colortable y
raster srtm_petar_25m_bilinear
range 50 1200
width 0.2
height 3.5
where 6.5 1.2
color black
nodata y
end

scale 1:150000
paper a4
end

end
```

³¹<http://en.wikipedia.org/wiki/PostScript>

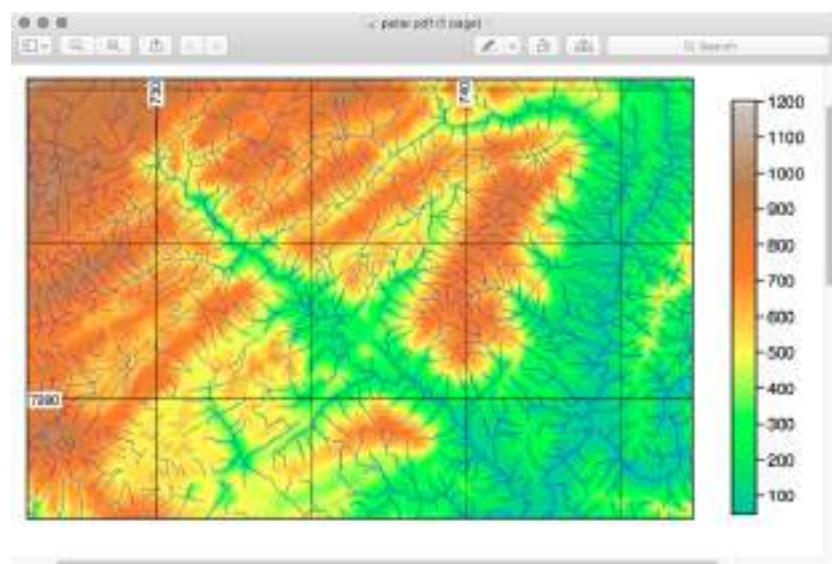


Figura 13.1. Mapa gerado pelo **ps.map**.

14. Automatizando tarefas com Python

Python é uma linguagem de programação bastante poderosa mas ao mesmo tempo de fácil aprendizado. Diversos programas de SIG a utilizam, tanto para automatizar tarefas através de *scripts* simples, como para desenvolvimento de novos módulos e *plug-ins*. Como o assunto é muito extenso, iremos discutir alguns aspectos do uso de Python com um *script* simples. Informações mais aprofundadas estão disponíveis na página https://grasswiki.osgeo.org/wiki/GRASS_and_Python.

```

# imports
import sys, os, csv
import math as m
import numpy as np
import matplotlib.pyplot as plt

import grass.script as grass
import grass.script.array as garray

# set overwrite
ow = True
# raster DEM
dem25m = 'srtm_petar_25m_bilinear'

# set region
grass.run_command('g.region', rast=dem25m, flags='pa')

# run r.param.scale
grass.run_command('r.param.scale', input=dem25m, output=dem25m + '_feature_21_0003', \
curvature_tolerance=0.0003, size=21, method='feature', overwrite=ow)

grass.run_command('r.param.scale', input=dem25m, output=dem25m + '_feature_21_0008', \
curvature_tolerance=0.0008, size=21, method='feature', overwrite=ow)

# list of features from r.param.scale
features = [1, 2, 3, 4, 5, 6]

# strings for curvature tolerance values
curv_tols = ['0003', '0008']

# loop through list of features
for feat in features:
    # loop through curvature tolerances
    for curv in curv_tols:
        # get name of raster map
        feat_map = dem25m + '_feature_21_' + curv
        print 'Elevation map: ' + dem25m
        print 'Stats for map: ' + feat_map
        print 'Feature class: ' + str(feat)
        # set mask
        grass.run_command('r.mask', raster=feat_map, maskcats=feat, quiet=True, overwrite=ow)
        grass.run_command('r.univar', map=dem25m, flags='t', separator='comma')

```

15. Referências Bibliográficas

- Bamler, R. & Hartl, P., 1998. Synthetic aperture radar interferometry. *Inverse Problems*, 14:R1–R54.
- Dassau, O., Holl, S., Neteler, M., & Redslob, M., 2005. *An introduction to the practical use of the Free Geographical Information System GRASS 6.0*. GDF Hannover bR.
- Demers, M. N., 2004. *Fundamentals of Geographic Information Systems*. John Wiley & Sons, New York.
- Eineder, M., Breit, H., Adam, N., Holzner, J., Suchandt, S., & Rabus, B., 2001. SRTM X-SAR calibration results. In *Geoscience and Remote Sensing Symposium, 2001. IGARSS '01. IEEE 2001 International*, volume 2, pages 748 –750 vol.2.
- Farr, T. G. & Kobrick, M., 2000. Shuttle Radar Topography Mission produces a wealth of data. *EOS (Transactions, American Geophysical Union)*, 81:583–585.
- Farr, T. G., Rosen, P. A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Seal, D., Shaffer, S., Shimada, J., Umland, J., Werner, M., Osokin, M., Burbank, D., & Alsdorf, D., 2007. The Shuttle Radar Topography Mission. *Review of Geophysics*, 45:RG2004.
- Grohmann, C. H., 2004. Morphometric analysis in Geographic Information Systems: applications of free software GRASS and R. *Computers & Geosciences*, 30:1055–1067.
- Grohmann, C. H. & Riccomini, C., 2009. Comparison of roving-window and search-window techniques for characterising landscape morphometry. *Computers & Geosciences*, 35:2164–2169.
- Grohmann, C. H., Smith, M. J., & Riccomini, C., 2010. Multiscale Analysis of Topographic Surface Roughness in the Midland Valley, Scotland. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(4):1200–1213.
- Hofierka, J., Parajka, J., Mitasova, H., & Mitas, L., 2002. Multivariate Interpolation of Precipitation Using Regularized Spline with Tension. *Transactions in GIS*, 6(2):135–150.
- Horn, B. K. P., 1981. Hill Shading and the Reflectance Map. *Proceedings of the IEEE*, 69:14–47.
- IBGE, 1998. *Noções básicas de Cartografia*. Diretoria de Geociências. Departamento de Cartografia.
- Jarvis, A., Reuter, H. I., Nelson, A., & Guevara, E., 2008. Hole-filled SRTM for the globe Version 4, available from the CGIAR-CSI SRTM 90m Database (<http://srtm.csi.cgiar.org>). last accessed: 11 March, 2013.
- Jet Propulsion Laboratory, 2014. U.S. Releases Enhanced Shuttle Land Elevation Data. Acessed Jan 15, 2015.
- Lillesand, T. M. & Kiefer, R. W., 1987. *Remote sensing and image interpretation*. John Wiley & Sons (2nd ed). New York.
- Liu, C. C., 1987. A geologia estrutural do Rio de Janeiro vista através de imagens MSS do Landsat. In *I Simpósio de Geologia Regional RJ-ES, SBG*, Anais, pages 164–188.
- LPDAAC – Land Processes Distributed Active Archive Center, 2013. SRTM Collection. Technical report, U.S. Geological Survey.
- Miranda, E. E., 2005. *Brasil em Relevo*. Embrapa Monitoramento por Satélite, Campinas. Documento eletrônico. <http://www.relevobr.cnpm.embrapa.br>.
- Mitasova, H. & Hofierka, J., 1993. Interpolation by regularized spline with tension:II. Application to terrain modeling and surface geometry analysis. *Mathematical Geology*, 25:657–669.

- Mitasova, H. & Mitas, L., 1993. Interpolation by regularized spline with tension:I Theory and implementation. *Mathematical Geology*, 25:641–655.
- Neteler, M. & Mitasova, H., 2008. *Open Source GIS : A GRASS GIS Approach, Third Edition (The International Series in Engineering and Computer Science)*, volume 773. Springer, New York.
- NGA – National Geospatial-Intelligence Agency, 2014. NGA releases high-resolution elevation data to public. Accessed Jan 15, 2015.
- Rabus, B., Eineder, M., Roth, R., & Bamler, R., 2003. The Shuttle Radar Topography Mission—a new class of digital elevation models acquired by spaceborne radar. *ISPRS Journal of Photogrammetry & Remote Sensing*, 57:241–262.
- Reuter, H. I., Nelson, A., & Jarvis, A., 2007. An evaluation of void-filling interpolation methods for SRTM data. *International Journal of Geographical Information Science*, 21:983–1008.
- Riccomini, C. & Crósta, A. P., 1988. Análise preliminar de lineamentos em imagens de sensores remotos aplicada à prospecção mineral na área dos granitóides Mandira, SP. *Bol. Inst. Geoc. USP. Série Científica*, 19:23–37.
- Rodriguez, E., Morris, C. H., & Belz, J. E., 2006. A global assessment of the SRTM performance. *Photogrammetric Engineering and Remote Sensing*, 72:249–260.
- Shapiro, M. & Westervelt, J., 1991. *r.mapcalc*: An Algebra for GIS and Image Processing. Technical report, U.S. Army Construction Engineering Research Laboratory.
- Shary, P. A., Sharaya, L. S., & Mitusov, A. V., 2002. Fundamental quantitative methods of land surface analysis. *Geoderma*, 107:1–32.
- Smith, M. & Wise, S., 2007. Problems of bias in mapping linear landforms from satellite imagery. *International Journal of Applied Earth Observations and Geoinformation*, 9(1):65–78.
- Smith, W. H. F. & Wessel, P., 1990. Gridding with continuous curvatures splines in tension. *Geophysics*, 55(3):293–305.
- Valeriano, M. M., Kuplich, T. M., Storino, M., Amaral, B. D., Mendes, J. N., & Lima, D. J., 2006. Modeling small watersheds in Brazilian Amazonia with shuttle radar topographic mission-90m data. *Computers & Geosciences*, 32:1169–1181.
- van Zyl, J. J., 2001. The Shuttle Radar Topography Mission (SRTM): a breakthrough in remote sensing of topography. *Acta Astronautica*, 48:559–565.
- Wood, J., 1996a. Scale-based characterization of Digital Elevation Models. In Parker, D., editor, *Innovations in GIS 3*, chapter 13. London: Taylor and Francis.
- Wood, J., 1996b. *The Geomorphological characterization of Digital Elevation Models*. PhD thesis, University of Leicester, U. K. , Department of Geography, PhD Thesis.
- Wood, J., 2009. Chapter 14 Geomorphometry in LandSurf. In Hengl, T. & Reuter, H. I., editors, *Geomorphometry: Concepts, Software, Applications*, volume 33 of *Developments in Soil Science*, pages 333–349. Elsevier, Amsterdam.
- Zebker, H. A. & Goldstein, R. M., 1986. Topographic mapping from interferometric synthetic aperture radar observations. *Journal of Geophysical Research*, 91:4993–4999.