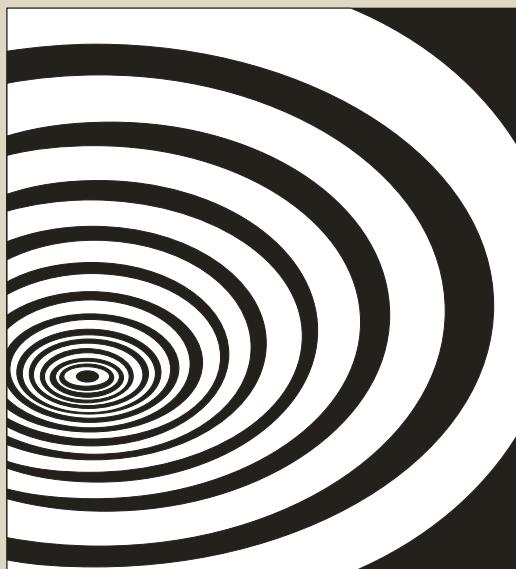


Modelagem de Dinâmica Ambiental com Dinamica EGO



**Britaldo S. Soares Filho, Hermann O. Rodrigues,
William L. Costa**

Tradução por Letícia Santos de Lima

**Centro de Sensoriamento Remoto/Universidade Federal de Minas Gerais. Av.
Antônio Carlos 6627, Belo Horizonte, 31270-901, MG, Brazil**

2009

Conteúdo

1. Introdução	4
1.1 Interface Gráfica do Dinamica EGO.....	6
1.2 Map Viewer	11
1.3 Formatos e Estrutura dos Dados.....	12
2. Lição 1: Construindo um modelo simples	14
3. Lição 2: Incorporando iterações a um modelo.....	24
4. Lição 3: Usando o conceito de região	30
5. Lição 4: Calculando uma superfície de custo acumulado e um caminho de menor custo ..	33
6. Lição 5: Avaliação Multicritério no Dinamica EGO	40
6.1 Primeiro Passo: Encontrando áreas sem impedimentos	40
6.2 Segundo Passo: Construindo buffers para definir áreas inapropriadas.....	42
6.3 Terceiro passo: Calculando tempo de deslocamento para cidades vizinhas	46
6.4 Quarto passo: Combinando os critérios booleanos.....	47
6.5 Quinto passo:	51
6.6 Sexto passo:.....	53
7. Lição 6: Construindo um modelo de simulação de mudanças de uso e cobertura da terra ..	57
7.1 Primeiro passo: Calculando matrizes de transição	59
7.2 Segundo passo: Calculando intervalos para categorização variáveis contínuas.....	62
7.3 Terceiro passo: Calculando os coeficientes dos Pesos de Evidência	67
7.4 Quarto passo: Analisando a correlação de mapas	69
7.5 Quinto passo: Ajustando e rodando um modelo de simulação LUCC	71
7.6 Sexto passo: Validando a simulação usando uma função de decaimento exponencial ...	74
7.7 Sétimo passo: Validando a simulação usando janelas múltiplas e função de decaimento exponencial	79
7.8 Oitavo passo: Executando simulação com formação de manchas	81
7.9 Nono passo: Executando a simulação com formação de manchas e expansão	83
7.10 Décimo passo: Projetando trajetórias de desmatamento	85
8. Linguagem de script e execução em console do Dinamica EGO	88
9. Recursos Avançados	93
9.1 Variando parâmetros em uma simulação	93
9.2 Usando sub-regiões em um modelo de simulação	95
9.3 Usando o tempo permanência.....	98
9.4 Usando o tempo de permanência e transições determinísticas.....	100
9.5 Usando saturação local	101
10. Métricas de Paisagem no Dinamica EGO	103

11. Estudo de caso REDD	104
11.1. Desenvolvendo um modelo de projeção econométrica da taxas de desmatamento baseado em mudanças no contexto socioeconômico dos municípios	107
11.2. Desenvolvendo um modelo de contabilidade de carbono	109
12. Referências	112
13. Lista de functores	114

1. Introdução

O que você irá aprender?

- O que é Dinamica EGO
- O que é um functor
- Sobre o que este guia prático cobre
- O que há de novo no Dinamica EGO versão 1.4 (Outubro, 2009)

Bem vindo ao Dinamica EGO. EGO é uma sigla para *Environment for Geoprocessing Objects* – Ambiente para Objetos Geoprocessáveis. A versão anterior do Dinamica foi totalmente reformulada para se tornar um ambiente de modelagem, agora com notáveis possibilidades para a construção de modelos, desde o mais simples modelo espacial estático até modelos dinâmicos bastante complexos, os quais podem envolver iterações aninhadas, retroalimentações dinâmicas, abordagem multiregiões, manipulação e combinação algébrica de dados em vários formatos, como mapas, tabelas, matrizes e constantes, processos de decisão para bifurcação e união de fluxos de execução, e uma série de algoritmos espaciais complexos para a análise e simulação de fenômenos espaço-temporais.

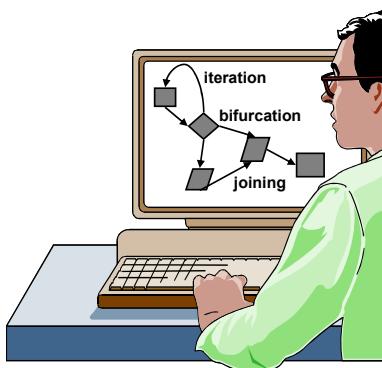


Fig. 1 – Um fluxo de dados, mostrando bifurcação e junção.

O ambiente do software, escrito em C++ e Java, contém uma série de algoritmos chamados *functores*. Cada functor executa uma operação. No presente, implementamos os algoritmos de análise espacial mais comuns disponíveis nos SIGs comerciais, além de uma série de algoritmos especialmente desenvolvidos para simulações espaciais, incluindo funções de transição e métodos de calibração e validação de modelos.

Esses functores são ligados em sequência para estabelecer um fluxo de dados na forma de grafos. Através da interface gráfica do Dinamica EGO o usuário pode criar modelos simplesmente selecionando, arrastando e conectando os functores através de seus portos, os quais representam conexões específicas para tipos de dados, como mapas, tabelas, matrizes, expressões matemáticas e constantes.

Functores podem ser agrupados em containers (recipientes), um tipo especial de functor que é usado, por exemplo, para executar iterações ou processar dados em regiões específicas de um mapa. Assim, os modelos podem ser construídos como se fossem diagramas e a execução segue um fluxo encadeado de dados. Essa interface amigável permite a criação de modelos espaciais, desde os mais simples até os mais complexos, os quais são salvos em uma linguagem de *script* no formato XML ou na linguagem de programação EGO.

Em suma, o software Dinamica EGO favorece simplicidade, flexibilidade e desempenho, otimizando velocidade e recursos computacionais, como memória virtual e processamento paralelo. A maioria de seus algoritmos é desenvolvida de forma a aproveitar arquitetura de múltiplos processadores. Além disso, Dinamica EGO suporta arquivos em formato raster de grande tamanho, até 64000 x 64000 células, usando paginação em disco. Por outro lado, se há memória disponível, o Dinamica pode carregar todos os mapas de entrada no início da execução do modelo e mantê-los na memória apenas enquanto eles são necessários. Dessa forma, o software acessa o disco apenas ao final de uma execução para guardar os resultados

finais, ou se o usuário especifica, ao final de uma iteração para salvar os mapas resultantes de cada iteração.

O objetivo desse guia prático é introduzir o usuário às inúmeras possibilidades do Dinamica EGO para a criação de modelos que podem, verdadeiramente, representar a complexidade de vários fenômenos geográficos.

O que há de novo na atual versão 1.4 (Outubro, 2009)

- ✓ Novo formato de script, EGO: O desempenho do parser (analisador) foi aprimorado para validar melhor os conteúdos do script e para mostrar mais mensagens de erro. Além disso, os formatos EGO e XML se tornaram 100% compatíveis, assim, nenhuma informação é perdida durante a conversão entre esses formatos.
- ✓ A linguagem de programação no formato EGO foi melhorada.
- ✓ Novo mecanismo de manuseio de mapas raster. Mesmo se não há memória virtual suficiente, o Dinamica EGO pode manipular grandes formatos raster usando swapping (troca) em disco para os mapas.
- ✓ Erros na leitura e escrita de conteúdos em alguns editores foram corrigidos.
- ✓ Paralelização de mais functores.
- ✓ Novo Visualizador de Mapa: suporta a visualização de grandes mapas e permite a manipulação de histograma.
- ✓ Agora você pode converter qualquer matriz de transição em tabelas e vice-versa.
- ✓ Os functores agora podem receber apelidos (alias). O alias representa uma variável de nome no script EGO, de forma que os modelos textuais se tornam muito mais comprehensíveis, e assim, mais fáceis de serem seguidos passo-a-passo.
- ✓ Novo mecanismo de atualização do Dinamica EGO via web.
- ✓ A opção Menu está agora com informação centralizada.
- ✓ Novo painel de propriedades através do qual você pode editar os portos de um functor.
- ✓ Verificador de ortografia para os comentários feitos dentro de um functor (apenas para o Inglês).
- ✓ Novo mecanismo de recuperação de dados que salva o modelo automaticamente, evitando a perda de informações.
- ✓ Agora o functor LookupTable tem uma função de auto-preenchimento e uma ferramenta de visualização de gráficos.
- ✓ Nova rotina de procura de functores por nome, alias (apelido) e comentários.
- ✓ Look&Feel personalizado.
- ✓ Utilitário para backup de modelos.
- ✓ Correções de erros e outras otimizações de desempenho.
- ✓ Painel de opções com propriedades do modelo. Uma descrição do modelo, incluindo autoria, versão, e outras informações, pode ser associada ao modelo. Essas propriedades do modelo ficarão visíveis como um tooltip (ferramenta de dica) no Windows Explorer.
- ✓ Novos operadores de álgebra de mapas, incluindo declividade, operadores hidrológicos como direção e acúmulo de fluxo, e uma função para estimar densidade de kernel de eventos pontuais. Functores para determinar e obter classes de mapas, e

- novos containers para extrair uma camada de um conjunto cubo raster e para reunir mapas em um raster multicamadas.
- ✓ Functores para controle de fluxos de execução renomeados para representar melhor seus processos.
- ✓ *Run external process* permite chamar um programa de dentro do Dinamica EGO.
- ✓ Operadores de vizinhança para mapas e novos operadores para tabelas foram incluídos no *Calculate map*.
- ✓ Um guia prático foi incluído, com lições sobre recursos avançados e também um estudo de caso REDD.
- ✓ Modelos para métricas de paisagem foram incluídos.
- ✓ Gráficos agora podem ser produzidos e salvos a partir de tabelas.
- ✓ Nomes de categorias e valores podem ser inseridos no cabeçalho das tabelas.
- ✓ Valores lineares podem ser interpolados em uma tabela.

1.1 Interface Gráfica do Dinamica EGO

O que você irá aprender?

- Interface gráfica do Dinamica EGO
- Ferramentas de interface

No Dinamica EGO, os modelos são construídos através de uma sequência de functores conectados através de portos de entrada e saída compatíveis. Assim, os dados fluem através desses operadores para produzir um resultado desejável que representa a solução de uma questão a respeito de algum aspecto do meio ambiente.

A interface gráfica do Dinamica EGO é dividida em 6 janelas. No sentido horário, a partir do topo à esquerda, são elas: 1) a janela Library (biblioteca), 2) o Sketch (plano de desenho), 3) o Message Log (janela de mensagem), 4) a janela Bird View (visão de pássaro), 5) a janela Functor Properties (propriedades dos functores), 6) as abas do Explorer (visão em árvore)(fig. 2).

Um modelo é construído selecionando e arrastando os functores a partir da janela Library para o plano de desenho, ou seja, a janela Sketch. A janela Bird View proporciona uma visão geral do modelo e é especialmente útil para grandes modelos que não se encaixam inteiramente no plano do sketch quando mostrados no tamanho padrão. A área sombreada na janela Bird View corresponde à área que a janela sketch abrange no momento. Movendo a área sombreada, você pode centralizar o sketch em uma parte específica do modelo. A visão em forma de árvore fornece uma compreensão hierárquica do modelo, permitindo ao usuário selecionar ou localizar um functor e, em sequência, editar suas propriedades através da janela Functor Property. Na parte mais baixa da interface está a janela Message Log, um espaço reservado para mensagens textuais que relatam a ocorrência de erros, alertas, informações e alertas de debug. Como padrão, o texto do Log irá mostrar o nível “informação”, mas o usuário pode mudar isso pressionando o botão para o nível desejado. **DICA:** Você pode aumentar o desempenho de alguns modelos reduzindo o nível da mensagem de log. Outro jeito de fazer isso é usando o functor *Log Policy*. O plano de sketch possui uma ferramenta de janela ao longo de sua borda superior. **DICA:** Você pode mudar a perspectiva da interface. Para isso, vá até a opção retrátil Window, presente na barra de menu, e escolha uma das três perspectivas disponíveis, através do item Perspective.

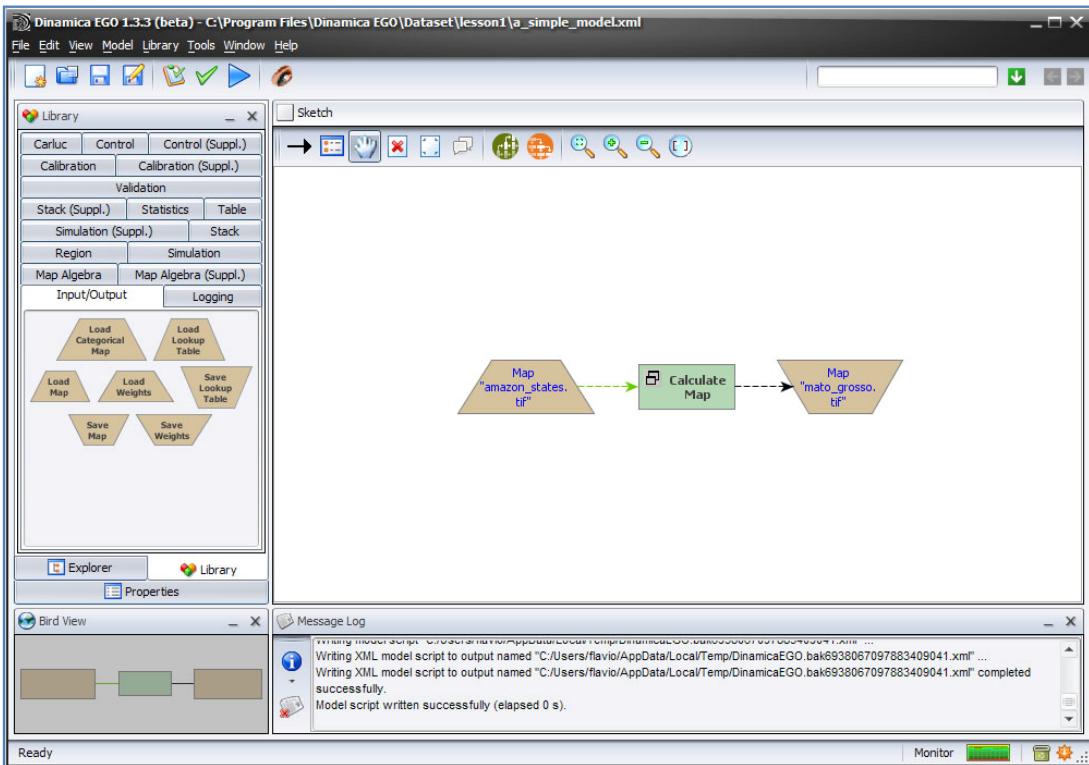
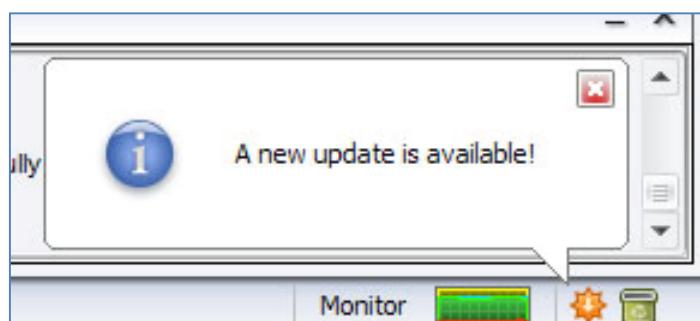


Fig. 2 – Interface Gráfica do Dinamica EGO

DICA: Você pode redimensionar ou fechar janelas. Isso pode se tornar útil à medida que o modelo cresce em número de functores. Você precisa apenas aumentar o tamanho da janela *sketch*, o que reduzirá automaticamente o tamanho das demais.

Na barra ao topo da janela do aplicativo, você também irá encontrar a janela do monitor e a ferramenta Garbage Collector (coletor de lixo), a qual permite que o usuário reduza a memória alocada dinamicamente pelo Dinamica EGO. Posicione o mouse sobre a janela de monitor para ampliá-la.

Quando uma nova versão se encontra disponível, uma mensagem irá aparecer no canto inferior direito da janela da interface. Vá para o sítio de internet do Dinamica EGO (www.csr.ufmg.br/dinamicaego) para atualizar a sua versão.



A caixa de ferramentas do sketch permite que o usuário faça conexões, edição, seleção e movimento de functores, edição de portos, redação de comentários em um functor, organização do layout do modelo e exibição do layout em diferentes zooms (fig. 3). **DICA:** você pode mover a caixa de ferramentas e posicioná-la verticalmente no lado esquerdo da janela Sketch. Você também pode acessar a caixa de ferramentas do plano de Sketch indo na opção Edit da barra de menu.

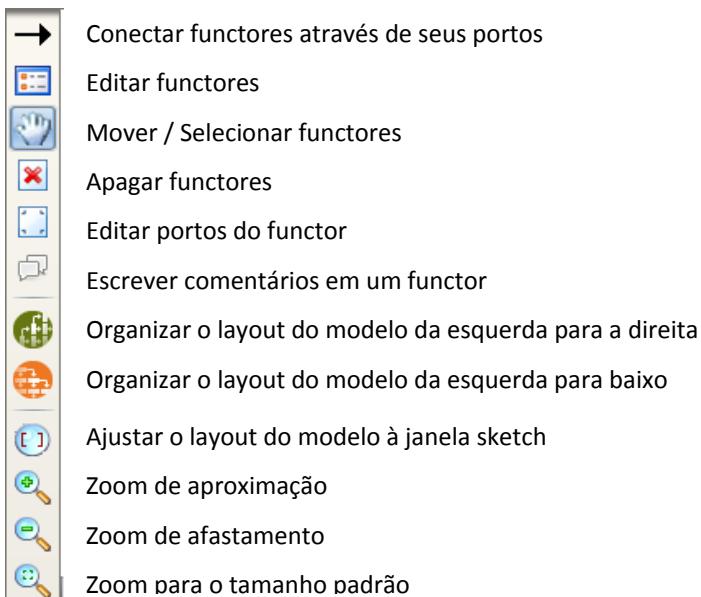


Fig. 3 – A caixa de ferramentas do plano de sketch.

DICA: você pode apagar functores selecionando-os com a ferramenta de mão e pressionando Delete.

Na janela Library você irá encontrar os functores disponíveis organizados em subconjuntos de acordo com seu tipo e aplicação. Os subconjuntos são:

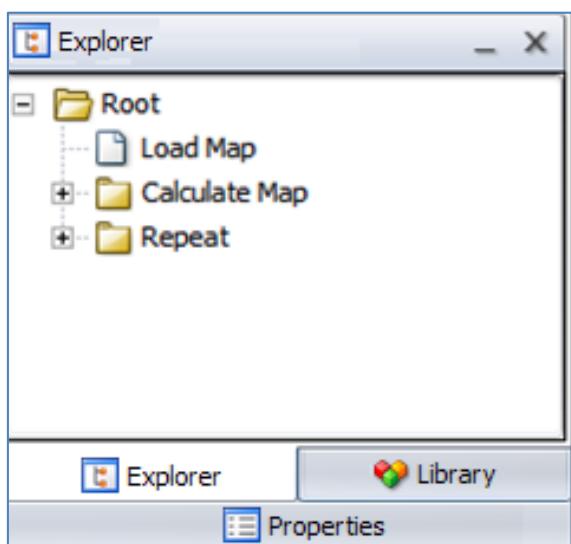
- 1) Map Algebra: um conjunto de algoritmos para análises espaciais.
- 2) Map Algebra supplementary: functores auxiliares para serem utilizados em conjunto com os functores do Map Algebra.
- 3) Region: esses functores permitem a divisão de um mapa em vários mapas regionais, cada um podendo ser usado separadamente em um submodelo específico ou iteração.
- 4) Simulation: conjunto de functores especialmente criados para desenvolver modelos espaço-temporais.
- 5) Simulation supplementary: functores auxiliares para serem usados em conjunto com functores do Simulation.
- 6) Stack: esses functores permitem a seleção de um parâmetro específico de acordo com o passo do modelo.
- 7) Stack supplementary: para ser usado em conjunto com os functores do Stack.
- 8) Statistics: functores para estatística espacial.
- 9) Table: um conjunto de functores para a manipulação de tabelas.
- 10) Validation: métricas para comparação de mapas.
- 11) Calibration: um conjunto de ferramentas para calibração de modelos espaço-temporais.
- 12) Calibration Supplementary: para ser usado em conjunto com os functores do Calibration.
- 13) Carluc: algoritmos do CARLUC, Modelo de carbono e mudanças de uso da terra (Hirsh et al., 2004). Este subconjunto está agora fora de uso, uma vez que, agora, o modelo CARLUC pode ser implementado diretamente usando os functores do Dinamica EGO.

- 14) Control: functores que controlam o encadeamento do fluxo de dados.
- 15) Control supplementary: para ser usado em conjunto com os functores do Control.
- 16) Input/Output: ferramentas para ler e salvar estruturas de dados na forma de mapas, tabelas e pesos de evidência.
- 17) Logging: específico para modelagem espacial da exploração da madeira (Merry et al., 2009).

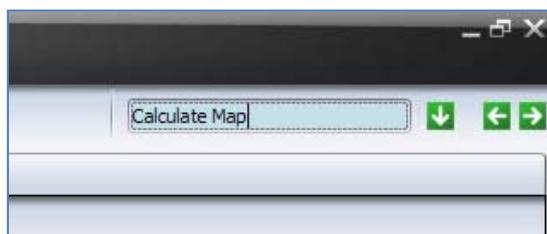
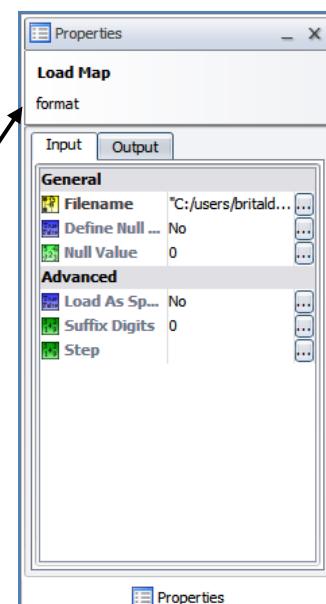
Por fim, os ícones na barra de menu são atalhos para criar, abrir, salvar um modelo, salvar um modelo em um novo arquivo, verificar a integridade do script de um modelo, rodar um modelo e abrir o visualizador de mapas. **DICA:** É possível abrir mais de uma janela da interface do Dinamica EGO. Isso pode ser útil, por exemplo, para a comparação da estrutura de dois modelos. Você também pode abrir uma série de janelas do visualizador de mapas para ver vários mapas ao mesmo tempo. Para isso, clique novamente no ícone olho.



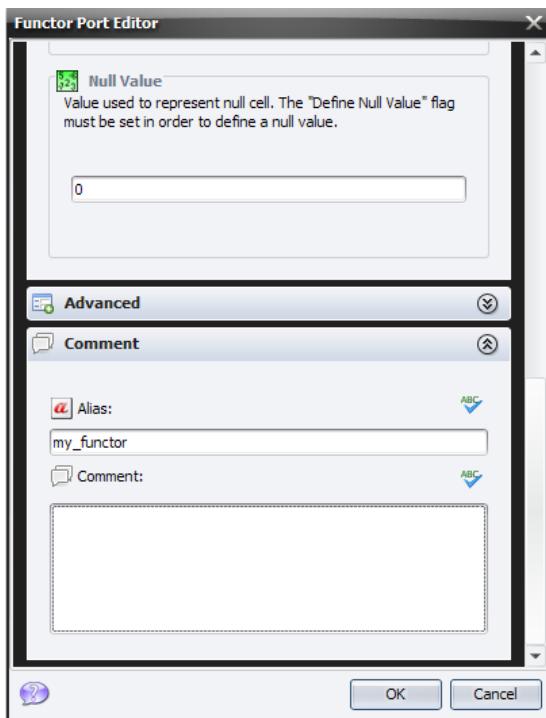
A partir do momento em que você começar a construir um modelo, você perceberá que o Dinamica EGO possui algumas feições que auxiliarão você a navegar através do modelo. Você pode usar a janela Explorer para ver uma representação em forma de árvore do seu modelo.



Clicando em um elemento da árvore, você irá selecioná-lo e habilitá-lo para edição através da janela Properties:

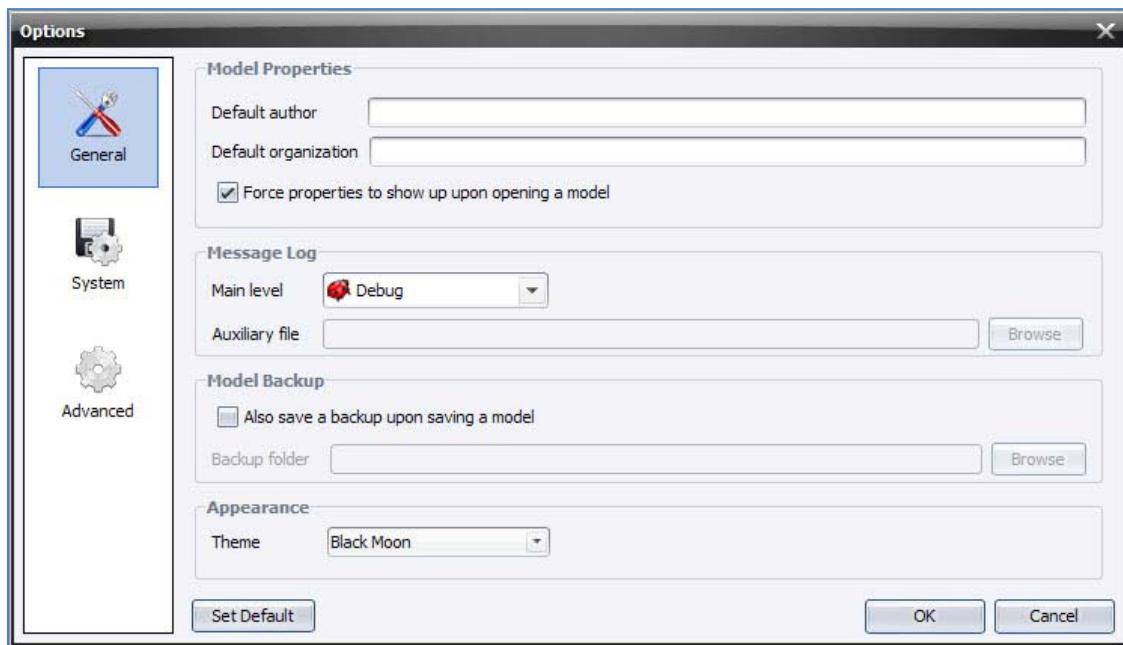


DICA: Você pode facilmente encontrar qualquer functor no modelo, digitando seu nome em uma caixa de texto localizada à direita no topo da interface. Você também pode iniciar uma procura pelo alias de um functor ou comentários.



Agora é possível dar um apelido (alias) a um functor de formar a facilitar a compreensão do modelo. Clique em um functor com a ferramenta de edição de functores, vá à aba Comment e altere a caixa alias.

Uma nova janela de opção está disponível para inserir informações sobre o modelo e para ajustar parâmetros do ambiente Dinamica EGO. Vá para Main Menu, Tools e Options:



A janela de opções é dividida em 3 abas. A primeira aba é para o preenchimento de informações do modelo, ajuste de mensagem de Log, backup do modelo, e Look&Feel. A segunda aba é para o ajuste dos parâmetros do sistema, tais como backup, auto-recuperação e pasta de arquivos temporários. Nas opções avançadas, a troca (swap) em disco do mapa pode ser desabilitada. **DICA:** Se você desabilitar o swap em disco do mapa, os mapas raster serão mantidos na memória virtual. Lembre-se que modelos com grande quantidade de dados podem causar “Allocation failed” (falha de alocação de memória) e consequente interrupção dele.

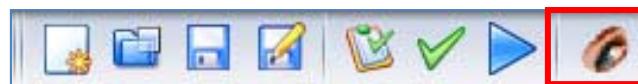
Para ver a lista de functores disponíveis no Dinamica EGO, use a lista Dump Functor, localizada no menu Help. Uma lista de functores será visualizada dentro da janela Log. Vá também ao Help Contents para obter descrições detalhadas de cada functor e seus algoritmos.

1.2 Map Viewer

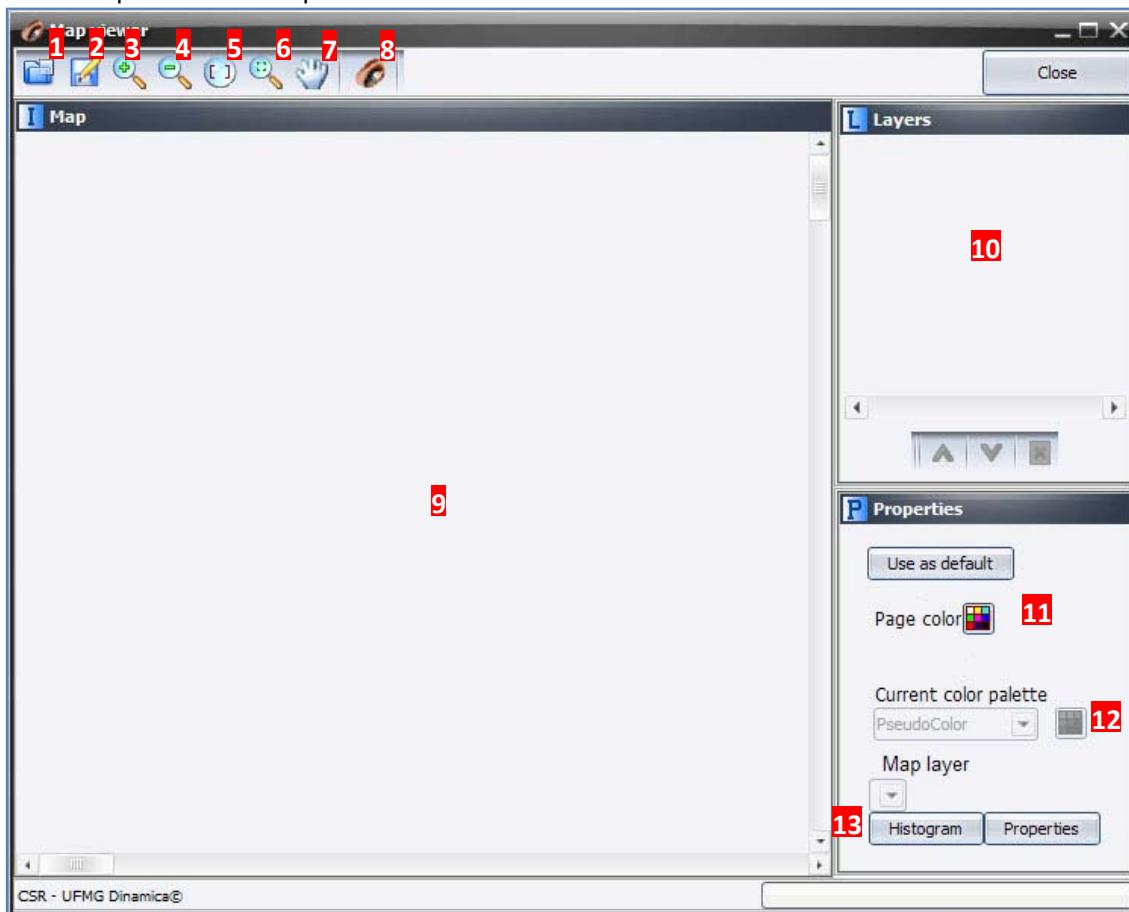
O que você irá aprender?

- Usando o Visualizador de Mapas

O Dinamica EGO disponibiliza uma ferramenta para visualizar os mapas de entrada e saída. Clique no Map Viewer na barra de ferramentas principal.

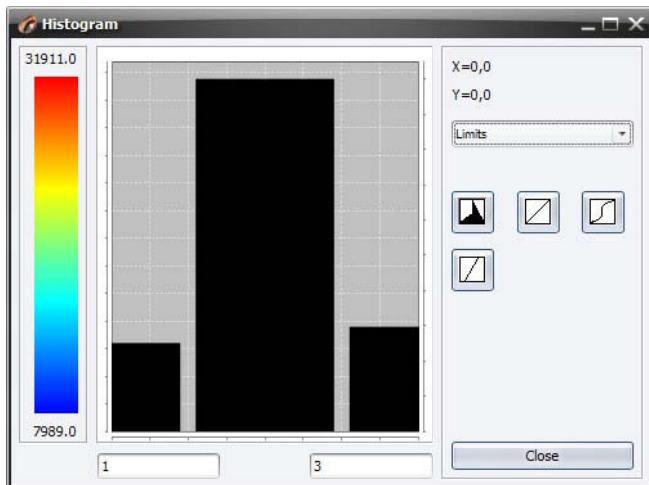


Esta é a aparência do Map Viewer:



- 1) Abrir um mapa
- 2) Salvar um mapa como um novo arquivo
- 3) Zoom de aproximação
- 4) Zoom de distanciamento
- 5) Zoom 1:1
- 6) Ajustar mapa à tela
- 7) Mover o mapa
- 8) Abrir nova janela do visualizador
- 9) Janela de mapa

- 10) Camadas: Lista de camadas para um cubo de mapas raster
- 11) Mudar cor do pano de fundo
- 12) Escolher paleta de cores para visualizar um mapa. **DICA:** você pode editar a paleta de cores.
- 13) Manipular histograma e mostrar propriedades do mapa



Para uma melhor visualização, Map Viewer permite alterar o histograma do mapa usando transformação linear e por equalização do histograma.

1.3 Formatos e Estrutura dos Dados

O que você irá aprender?

- Estrutura de dados no Dinamica EGO
- Formatos de arquivo no Dinamica EGO

Dinamica EGO suporta dados em vários formatos, incluindo mapas raster ou imagens, tabelas, matrizes, e arquivo de coeficientes de Pesos de Evidência.

Para dados espaciais, **Dinamica EGO suporta apenas conjuntos de dados raster**. Assim, você precisará preparar o seu conjunto de dados em um pacote SIG e, em seguida, exportar os mapas em um dos três formatos especificados abaixo. Embora o georreferenciamento seja suportado e necessário, todos os conjuntos de dados raster em um modelo devem ter o mesmo número de colunas e linhas. Além disso, eles devem estar ancorados em um mesmo ponto de registro com as mesmas coordenadas no espaço geográfico (fig. 4).

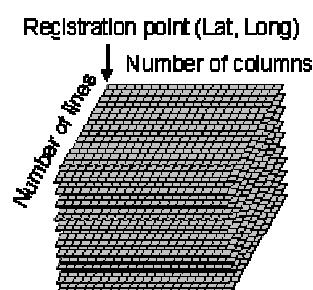


Fig.4 a cube raster dataset

Dinamica EGO lê e escreve dados raster em três formatos: formato ER Mapper, Geotiff, e ArcView ASCII. Os sistemas de coordenadas suportados para transformação para o formato ERMapper são o Geodético (WGS 84, SAD69, Córrego Alegre) e UTM (WGS84, SAD 69, Córrego Alegre). Quando um sistema de coordenadas não é suportado, assume-se a projeção LOCAL e datum WGS84. Para ASCII/ArcView, são sempre assumidos a projeção LOCAL e o datum WGS84. Todos os sistemas de coordenadas Geotiff são suportados se uma conversão não for

necessária. Quando a dimensão de célula Geotiff não é encontrada, uma resolução de 100 metros é assumida. O formato Geotiff usando tiling (ladrilhamento) é também suportado, assim, você não encontrará problemas ao importar e exportar seu conjunto de dados através dos SIGs mais comuns, tais como IDRISI, SPRING ou ARCGIS 9.*.

DICA: Encontre mais informações a respeito do Geotiff em <http://trac.osgeo.org/geotiff/>

Na álgebra de mapas, o conceito de valor nulo é muito importante para se obter um resultado coerente da execução do modelo. Nulo significa ausência de dados. Assim, um mapa que contém uma área irregular de interesse, a qual não cobre completamente o plano geográfico, deverá ter uma representação para as células nulas. O valor reservado para a representação da célula nula pode variar dependendo do tipo de dado, isto é, o tamanho utilizado, em termos de bits, para armazenar os valores das células do mapa. Dinamica EGO suporta os seguintes tipos de dados:

1-Bit Integer [0, 1]
Signed 8 Bit Integer [-128, 127]
Unsigned 8 Bit Integer [0, 255]
Signed 16 Bit Integer [-32768, 32767]
Unsigned 16 Bit Integer [0, 65535]
Signed 32 Bit Integer [-2147483648, 2147483647]
Unsigned 32 Bit Integer [0, 4294967295]
IEEE 754 32 Bit Real [-3 4028235E38, 3 4028235E38]

Normalmente, o menor valor negativo é usado para representar o valor nulo. Por exemplo, o valor -32768 é usado para dados do tipo “Signed 16 Bit Integer”. **DICA:** Escolha sempre o tipo de dado capaz de cobrir todo o espectro de valores contidos nas variáveis a serem representadas. Por exemplo: no caso da altitude, variando de -10 metros a 4000 metros, o tipo de dado a ser utilizado deverá ser o “Signed 16 Bit Integer” ou “IEEE 754 32 Bit Real”.

DICA: Em alguns casos, você precisará (e deverá) definir o valor nulo ao carregar um conjunto de dados Geotiff, o qual não possui campo para essa definição (altamente recomendado). Na lição 1, você aprenderá como fazer isso.

As tabelas são um modo conveniente de representar dados de atributos, normalmente pertencentes a alguma zona geográfica, por exemplo: país, estado ou município. Dinamica EGO pode ler dados no formato Comma Separated Value (valores separados por vírgula), no qual a primeira coluna representa a chave e a segunda, o valor, como a seguir:

DICA: A primeira linha da tabela deve conter o título da coluna “Key” com o nome da variável, tal como população, países, etc.

Key	Value
1	10
2	30
3	15

Uma matriz de transição é também armazenada usando esse formato; a única diferença é que a chave emprega um algarismo composto para representar a transição, como no exemplo abaixo:

Key	Value
1.002	0.223567
1.003	0.379618
2.001	0.024841
2.003	0.030573
3.002	0.000348

Assim, a tabela acima é equivalente à seguinte matriz de transição: Observe que os valores na diagonal não necessitam ser preenchidos, também não é necessário expressar transições iguais zero.

	1	2	3
1	-	0.223567	0.379618
2	0.024841	-	0.030573
3	0	0	-

Outro formato suportado é o arquivo Peso de Evidências – um arquivo texto contendo os coeficientes dos Pesos de Evidências. Este arquivo é obtido através do método dos Pesos de Evidência usado no processo de calibração. **DICA:** Você pode editar este arquivo diretamente em um editor de texto ou usando o editor gráfico dos Pesos de Evidência ligado ao functor *Load Weights*.

2. Lição 1: Construindo um modelo simples

O que você irá aprender?

- Functores e portos
- Como conectar functores
- Escrevendo expressões algébricas e lógicas
-
- Functores:
 - *Load Map*
 - *Save Map*
 - *Calculate Map*
 - *Number Map*

Uma solução para um modelo espacial envolve a seleção de mapas de um modelo cartográfico – que é um cubo de mapas raster registrados, no qual cada mapa se refere a uma característica geográfica particular –, processamento através de uma sequência ordenada, e então armazenar os resultados intermediários para processamentos subsequentes (fig. 5). Para melhorar o desempenho, o sistema de processamento do Dinamica EGO acessa o disco somente uma vez para ler os mapas de entrada, se memória suficiente estiver disponível, e depois, para escrever os resultados finais (fig. 6). Assim, um modelo no Dinamica EGO é representado por uma sequência de functores, cuja execução acontece da esquerda para a direita. Um modelo simples consiste em um functor que carrega o dado, por exemplo, um mapa, outro functor que executa um cálculo, e um terceiro que escreve o resultado em um arquivo. Note que os functores são conectados uns aos outros através de setas. Para fazer isso, eles devem trocar dados compatíveis através de suas entradas e saídas. As entradas e saídas em functores são chamadas de portos no Dinamica EGO. Consequentemente, os functores são conectados através de portos de entradas e saídas compatíveis. Vamos começar construindo o seu primeiro modelo no Dinamica EGO.

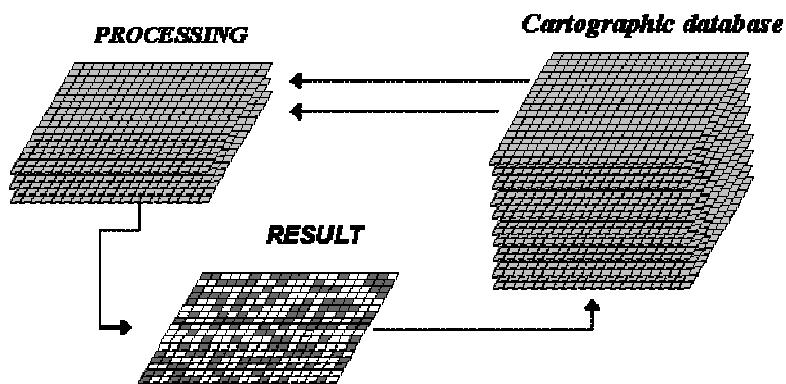


Fig. 5. Uma sequência tradicional de processamento de mapas.

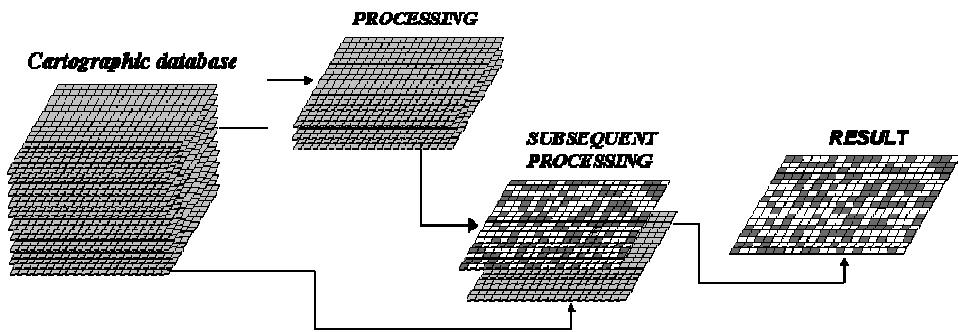
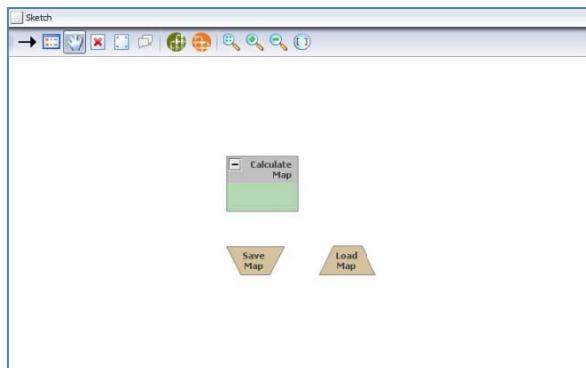
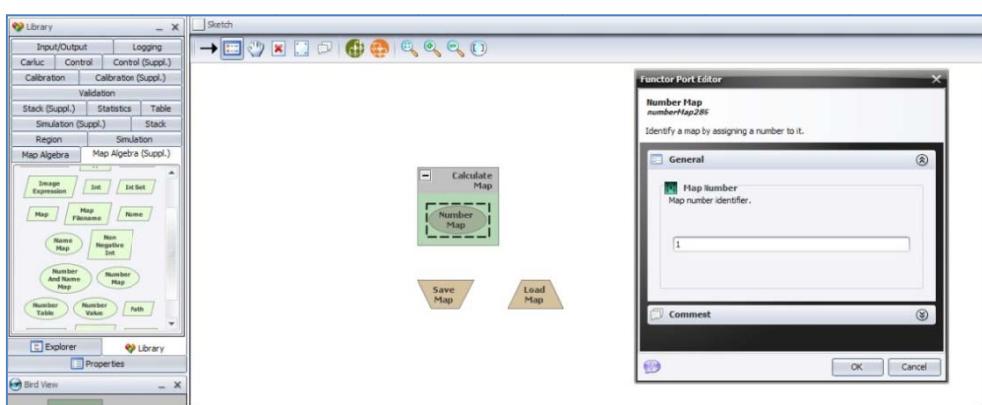


Fig. 6. Sistema de processamento do Dinamica EGO, nesse caso não é necessário salvar resultados intermediários nos arquivos em disco.

A partir da janela Library, pegue e arraste até o plano de sketch os functores *Load Map*¹ e *Save Map*, os quais estão localizados na aba Input/Output. Agora arraste o container *Calculate Map* da aba Map Algebra. Lembre-se que um container é um tipo especial de functor que pode conter functores suplementares ou mesmo uma sequência de functores. Os containeres diferem de um functor pela barra de título. **DICA:** Ao clicar na parte superior esquerda do ícone do container, você poderá abri-lo e fechá-lo. Você verá algo semelhante ao seguinte:



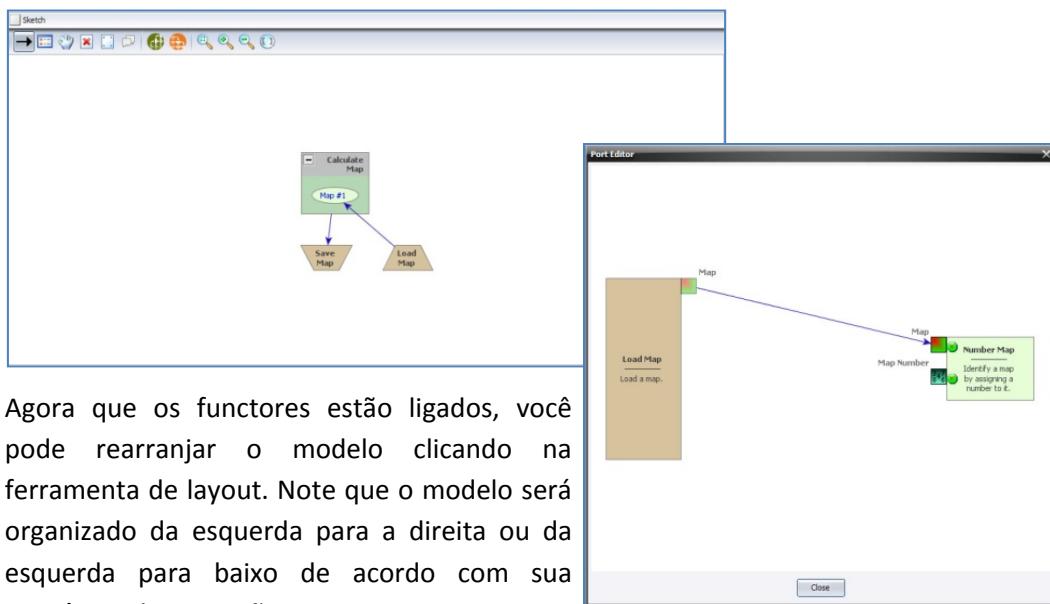
Agora você precisa conectar os functores para criar o modelo. Primeiro, vamos entender o que faz *Calculate Map*. Esse container é uma calculadora usada para álgebra de mapas, com o objetivo de combinar e processar mapas, tabelas e constantes. Como um container, ele não funciona por si mesmo. É necessário adicionar functores suplementares a ele. Nesse caso, cada mapa processado por esse container será representado pelo functor *Number Map*, disponível na aba Map Algebra Supplementary. Arraste o *Number Map* para dentro do container *Calculate Map*. O container irá reajustar seu tamanho para acomodar o *Number Map*.



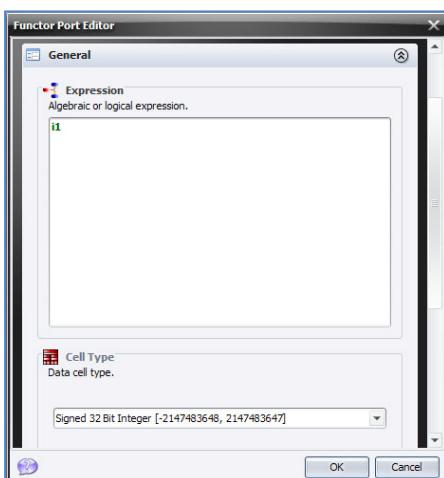
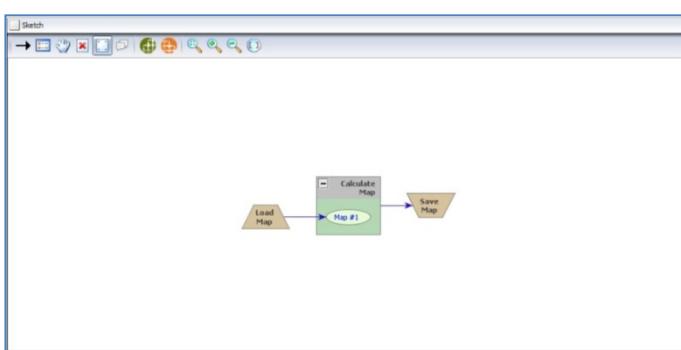
¹ Todos os nomes dos functores e containers são representados em *italíco* neste tutorial. Opções do modelo e parâmetros internos são representados em **negrito** e dados a serem inseridos são representados entre aspas.

Clique no *Number Map* com a ferramenta de edição de functores, *Edit Functor*, e insira “1” (não escreva as aspas). Este é um número identificador de um mapa e será representado dentro da caixa de equações como **i1** (input 1). Você pode inserir outros mapas adicionando mais functores *Number Map*, mas cada um deve ter um único número identificador. Agora você pode conectar o functor *Load Map* ao *Number Map* e o container *Calculate Map* ao *Save Map*. Use a ferramenta de conexão (o ícone de seta) para estabelecer as conexões. Observe que a conexão é feita automaticamente porque há apenas uma opção para portos compatíveis nos respectivos functores e no container.

Clicando na seta com a ferramenta de edição de portos, *Edit Functor Ports*, você poderá visualizar as ligações entre um par de functores.



Agora que os functores estão ligados, você pode rearranjar o modelo clicando na ferramenta de layout. Note que o modelo será organizado da esquerda para a direita ou da esquerda para baixo de acordo com sua sequência de execução.



No próximo passo, abra o container *Calculate Map* clicando sobre *Edit Functor* e, em seguida, sobre *Calculate Map*. Observe que o mapa #1 é representado por **i1**. *Calculate Map* possibilita a formulação de várias equações algébricas e lógicas contendo mapas, tabelas e constantes. A

tabela a seguir apresenta exemplos de operadores que podem ser aplicados para processar dados neste container. A tabela é dividida em 4 grupos: lógico, matemático, operadores de tabela e operadores de vizinhança. Além de escrever a equação, há dois parâmetros que precisam ser ajustados. O tipo de célula do mapa de saída e o valor nulo. Você sempre irá encontrar estes parâmetros em functores que produzem mapas como saída. O padrão é **Signed 32 Bit Integer**, mas você pode usar **IEEE 32 Bit Real** para representar números fracionários. **DICA:** Tente usar

sempre a representação mais econômica para o tipo de célula a fim de economizar memória. Se você não está seguro a respeito da faixa numérica dos dados de saída, use uma representação de número real.

Operadores Lógicos

Operador	Descrição	Símbolo	Exemplo de Uso
Boolean Or (Booleano ou)		or 	notisNull(i1) or isNull(i2) notisNull(i1) isNull(i2)
Boolean And (Booleano e)		and &&	notisNull(i1) and isNull(i2) notisNull(i1) && isNull(i2)
Equal (Igual)		= ==	i1 = 2 i1 == 2
Not Equal (Não igual)		!= /= <>	i1 != 2 i1 /= 2 i1 <> 2
Greater Than (Maior que)		>	i1 > 2
Greater Than Or Equal (Maior que ou igual)		>=	i1 >= 2
Less Than (Menor que)		<	i1 < i2
Less Than Or Equal (Menor que ou igual)		<=	i1 <= i2
Conditional (Condisional)	Executa o segundo ou terceiro termo da equação tendo como condição o primeiro.	if then else	if notisNull(i1) and isNull(i2) then i3 else if isNull(i1) then i1 – i1 / i2 else (i1 / i2) ? (i1 – i2)

Operadores Matemáticos

Operator	Description	Symbol	Usage Example
Add (Adicionar)		+	i1 + i2
Subtract (Subtrair)		-	i1 – i1 / i2
Times (Multiplicar)		*	i1 * i2
Divide (Dividir)		/	i1 / i2
Mod (Resto)		%	i1 % 100
Power (Potência)		^	i1 ^ 3
Catch Error (Capturar erro)	Deteta um erro algébrico e o substitue pelo resultado de outra expressão	?	(i1 / i2) ? (i1 – i2)
Value (Valor)			2 + i1 / -3.5e-2

Get Variable Value (Obter valor da variável)	Retorna o valor de uma variável	vX where X is an integer value from 1 to 100	v1 + t1[v2 + 4]
Get Image Value (Obter valor da imagem)		iX where X is an integer value from 1 to 100	i2
Get Image Value At Location (Obter valor da imagem na coordenada)	Retorna o valor de uma imagem em uma coordenada específica de célula	iX[,] where X is an integer value from 1 to 100	i1[line - 1, column - 2]
Get Image Null Value (Obter valor nulo da imagem)	Retorna o valor nulo da célula atual	null null(iX)	if i1 > 2 then i1 else null if null(i2) > 2 then 1 else null
Is Null (É nulo)		isNull(iX) where X is an integer value from 1 to 100	if not isNull(i1) then i1 else i2
Boolean Not (Não Booleano)		! not	not isNull(i1) ! isNull(i1)
Get Line Number (Obter número da linha)	Retorna o valor da linha da célula atual	line	line + 1
Get Column Number (Obter número da coluna)	Retorna o número da coluna da célula atual	column	if column / 2 > 50 then 1 else null
Random (Aleatório)	Gera um valor aleatório usando a distribuição de probabilidade uniforme	rand	if rand > 0.5 then i1 else i2
Negate (Negativo)		-	- ceil(i1 + i2)
Squared Root (Raiz quadrada)		sqrt()	sqrt(i1 / i4)
Sin (Seno)		sin()	sin(i1 / i4)
Cos (Cosseno)		cos()	cos(i1 + i2)
Tan (Tangente)		tan()	tan(i1 * i5 + 6)
Acos (Arco-cosseno)		acos()	acos(i1 + i2)
Asin (Arco-seno)		asin()	asin(i1 + i2)
Atan (Arco-tangente)		atan()	atan(i1 + i2)
Ceil (Teto)		ceil()	ceil(i1 + i2)
Exp (Exponencial)		exp()	exp(i1[i1 + i2])
Floor (Piso)		floor()	floor(i1 + i2)
Round (Arredondar)		round()	round(i1 / i4)
Abs (Valor absoluto)		abs()	abs(i1 + i2)
Ln (logaritmo neperiano)		ln()	ln(i1 / i4)
Log (logaritmo na base 10)		log()	log(i1 / i4)
Max (Maior valor)		max()	max(i1, i2)
Min (Menor valor)		min()	min(i1, i4)
Signal (Sinal)	Retorna +1, se a	signal	signal(i1 - 4)

	expressão for positiva, -1, se for negativa, e 0, se não for nenhum dos dois casos		
Abort (Abortar)	Aborta a execução do modelo	abort	if i1 > 0 then i1 * i2 + 4 else abort

Operadores de Tabela

Operador	Descrição	Símbolo	Exemplo de Uso
Operadores de Tabela	Operadores de tabela retornam o valor correspondente a uma dada chave de acordo com uma regra. Usa-se a seguinte sintaxe: tX[N] onde: X é o identificador de tabela; N é o operador de regra.		
Get Table Value (Obter Valor da Tabela)	Retorna o valor na posição da chave X da tabela	tX[] onde X é um valor inteiro entre 1 e 100	t2[i1 + 2]
Get Table Equal Lower Value (Obter valor da tabela igual ou menor que o valor limite)	Retorna o valor da tabela cuja chave possui o maior valor menor ou igual ao valor da chave de posição X da tabela.	tX[<=] tX{ } onde X é um valor inteiro entre 1 e 100	t2[<= 14] t2{14}
Get Table Lower Bound Value (Obter valor do limite inferior)	Retorna o valor da tabela cuja chave possui o maior valor menor que o valor da chave de posição X da tabela.	tX[<] onde X é um valor inteiro entre 1 e 100	t2[< i1 + 2]
Get Table Equal Upper Bound Value (Obter valor igual ao limite superior da tabela)	Retorna o valor da tabela cuja chave possui o menor valor maior ou igual ao valor da chave de posição X da tabela.	tX[>=] onde X é um valor inteiro entre 1 e 100	t2[>= i1 + i3]
Get Table Upper Bound Value (Obter valor do limite superior)	Retorna o valor da tabela cuja chave possui o menor valor maior que a chave de posição X da tabela.	tX[>] onde X é um valor inteiro entre 1 e 100	t2[> i7]
Get Table Closest Value (Obter valor mais próximo)	Retorna o valor da tabela cuja chave possui o valor mais próximo à chave de posição X da tabela.	tX[><] onde X é um valor inteiro entre 1 e 100	t2[>< 3 + i7]
Get Table Interpolated Value (Obter valor interpolado da tabela)	Retorna um valor interpolado linearmente usando as chaves vizinhas à posição X da tabela.	tX[/] onde X é um valor inteiro entre 1 e 100	t2[/ i2]

Operadores de Vizinhança

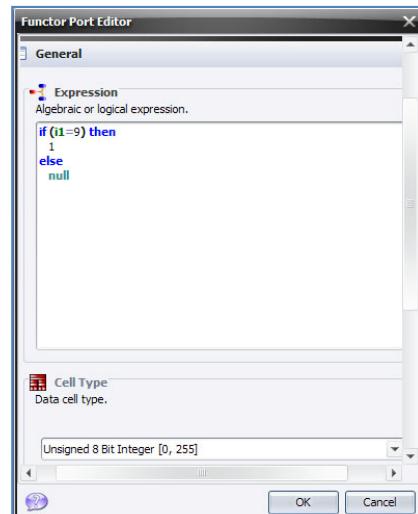
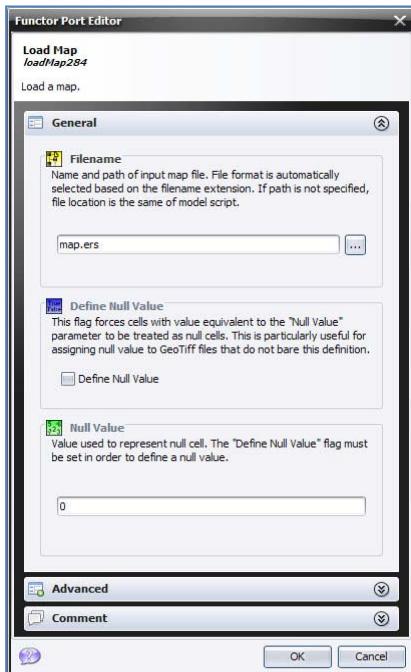
Operador	Descrição	Símbolo	Exemplo de Uso
Operadores de Vizinhança	<p>Operadores de vizinhança retornam o valor de uma operação dentro de uma janela definida de células vizinhas.</p> <p>Usa-se a seguinte sintaxe:</p> $\text{nbN}(iX, h, w, y, x)$ <p>onde:</p> <ul style="list-style-type: none"> N é o nome do operador; X é um identificador de imagem; h é o número de linhas da janela; w é o número de colunas da janela; y é a linha onde o centro da janela está ancorado na imagem; x é a coluna onde o centro da janela está ancorado na imagem. <p>O cálculo normalmente considera o centro da janela.</p> <p>Janelas com números pares de linhas e/ou colunas tem o centro deslocado em direção ao canto superior esquerdo.</p> <p>A âncora da janela (y e x) pode ser omitida quando o centro da janela está ancorado na linha e coluna atual. A sintaxe simplificada é a seguinte:</p> $\text{nbN}(iX, h, w)$		
Neighborhood Min (Vizinho Menor)	Retorna o menor valor entre as células vizinhas não nulas.	nbMin()	nbMin(i4, 2, 3, line-1, column)
Neighborhood Max (Vizinho Maior)	Retorna o maior valor entre as células vizinhas não nulas.	nbMax()	nbMax(i1, 4, 4) - 1
Neighborhood Sum (Somatório da Vizinhança)	Retorna a soma dos valores das células vizinhas não nulas.	nbSum()	nbSum(i3, 5, 5) + 7
Neighborhood Product (Produto da Vizinhança)	Retorna o produto dos valores das células vizinhas não nulas.	nbProd()	if not isNaN(i1) then nbProd(i1, 2, 2, 0, column) else 0
Neighborhood Count (Número de Vizinhos)	Retorna o número de células vizinhas não nulas.	nbCount()	nbCount(i2, 3, 3) + nbCount(i1, 3, 3, line, column+3)
Neighborhood Average (Média da Vizinhança)	Retorna a média aritmética dos valores das células vizinhas não nulas.	nbAverage()	round(nbAverage(i1, 7, 7))
Neighborhood Median (Mediana da Vizinhança)	Retorna a mediana dos valores das células vizinhas não nulas. Para um número ímpar de valores, o maior valor entre os dois valores medianos é considerado.	nbMedian()	nbMedian(i1, 5, 5)
Neighborhood Mode (Moda da Vizinhança)	Retorna a moda dos valores das células vizinhas não nulas. Se não há moda, um valor nulo é inserido. Se há mais de uma moda, a menor delas é considerada.	nbMode()	nbMode(i1, 5, 5)

Neighborhood Variance (Variância da Vizinhança)	Retorna a variância dos valores das células vizinhas não nulas de acordo com a expressão: $s^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$ onde X_1, X_2, \dots, X_n são as células vizinhas; \bar{X} é a média das células vizinhas.	nbVar()	nbVar(i4, 7, 7) / 25
Neighborhood Standard Deviation (Desvio Padrão da Vizinhança)	Retorna o desvio padrão dos valores das células não-nulas de acordo com a expressão: $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$ onde X_1, X_2, \dots, X_n são os valores das células vizinhas; \bar{X} é a média das células vizinhas.	nbStdDev()	nbStdDev(i2, 3, 3, line-1, column) + nbStdDev(i2, 3, 3) + nbStdDev(i2, 3, 3, line+1, column)

Agora que você aprendeu as vastas possibilidades desse container, vamos escrever uma equação simples:

if i1 = 9 then 1 else null

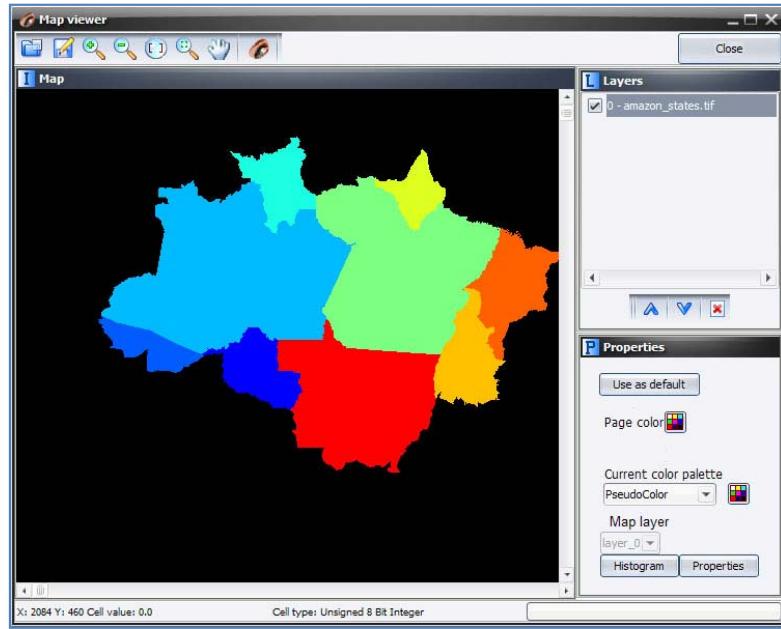
Significa: encontre a classe do mapa igual a 9, atribua às suas células o valor 1 e valor nulo a todas as outras. Mude o **Data cell type** para **Unsigned 8 Bit Integer** e o **Null Value** para “0”. Agora você pode fechar o container e abrir um arquivo de mapa para usar como dado de entrada para o modelo.



Abra o functor *Load Map* e carregue o arquivo “amazon_states.tif” da pasta **lesson1** (não escreva as aspas). Não se preocupe com as demais opções, pois elas não serão necessárias por agora.

DICA: Embora você não vá precisar disso nesta lição, você poderá ajustar o valor nulo marcando a opção **Define Null Value** e depois ajustando seu valor para “0” no campo **Null Value**.

Abra Map Viewer para visualizar o mapa. Pressione em “Histogram” e depois mude “Limits” para “actual” e certifique-se de que a paleta de cores, “Current Color Palette”, ativado é o **PseudoColor**. O mapa parecerá com este:



DICA: “Limits to actual” espalha o histograma do mapa para abranger toda a extensão de valores visualizáveis, i.e. de 0 a 255. Use preferencialmente **PseudoColor** ou **GrayScale** para dados quantitativos, e **Categorical** para dados classificados.

Agora você precisa salvar o resultado em um mapa. Abra *Save Map* e escreva “Mato_Grosso_state”.

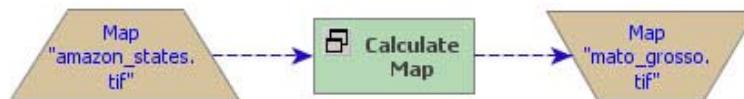


Na aba Advanced há uma opção para incluir sufixos no nome do arquivo, chamada **Suffix Digits**. Embora o padrão seja 2, mudaremos para zero, neste caso, uma vez que só haverá uma iteração. A opção **Use Compression** é aplicável apenas ao formato geotiff.

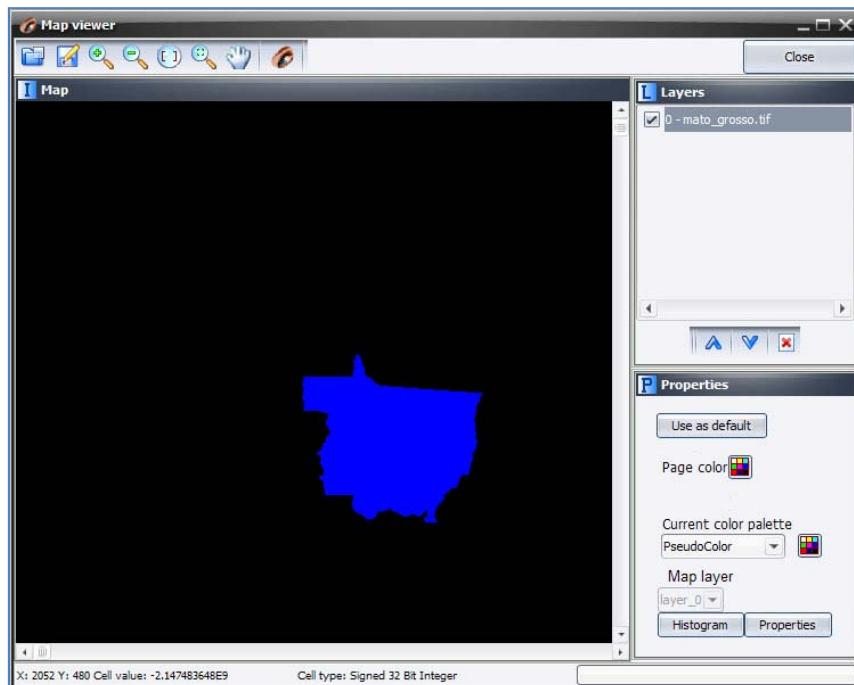
O mapa de entrada é um mapa categórico; seus valores não representam quantidades, mas sim, identificadores de classes do mapa e, neste exemplo, as classes são as seguintes:

Chave	Estado
1	Rondônia
2	Acre
3	Amazonas
4	Roraima
5	Para
6	Amapá
7	Tocantins
8	Maranhão
9	Mato Grosso

Feche o container *Calculate Map* clicando em sua barra. O modelo deverá se parecer com o seguinte:



Salve o modelo como “my_model” na pasta **lesson1**, clique em **Check Model Script Integrity** para verificar se o modelo está pronto para rodar e, se estiver, execute o modelo clicando no botão **Run Model Script** na barra de ferramentas superior. Em seguida, abra o resultado no Map Viewer. O mapa de saída irá representar apenas o estado do Mato Grosso.



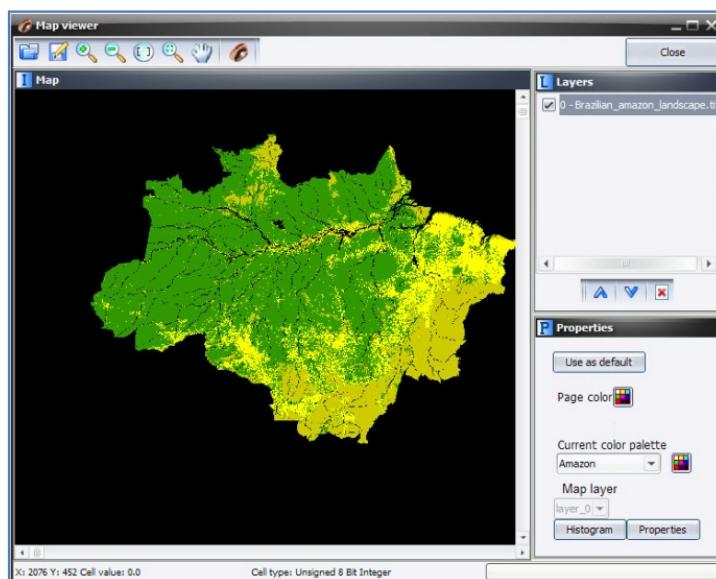
Parabéns, você concluiu com sucesso a lição 1. Agora vamos passar para a lição 2.

3. Lição 2: Incorporando iterações a um modelo

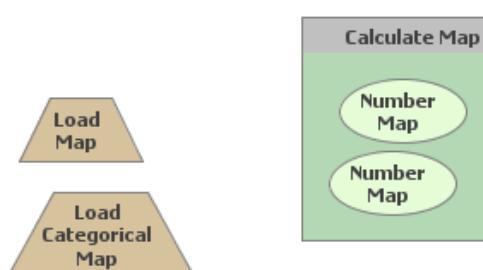
O que você irá aprender?

- Interações
- Usando Register Viewer
- Functores:
 - *Repeat*
 - *Step*
 - *Extract Map Attributes*
 - *Load Categorical Map*
 - *Calculate Value*
 - *Mux Lookup Table*
 - *Set Lookup Table Values*

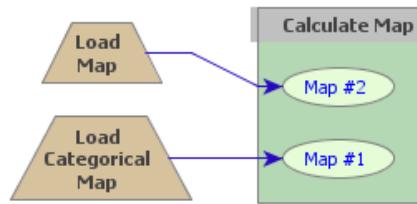
Abra o arquivo “brazilian_amazon_ landscape.tif” localizado na pasta **lesson2** no Map Viewer. Use **Amazon** como sua paleta de cores. Este é um mapa de cobertura da terra da Amazônia Brasileira. O mapa tem as seguintes classes: 1 – desmatado (amarelo), 2 – floresta (verde), e 3 – cerrado (marrom claro). A intenção deste exercício é calcular a área de floresta remanescente em cada estado.



Comece trazendo para o sketch um *Load Map* e um *Load Categorical Map*. Este último functor irá categorizar um mapa, caso este já não esteja categorizado, i.e. fará com que os valores das células sejam tratados como identificadores de classes do mapa, de forma que quando um mapa é carregado, é feita uma varredura no mapa para identificar todos os valores de células que representam diferentes categorias ou classes de mapas, produzindo como resultado uma lista de classes, a qual se torna encapsulada no cabeçalho do mapa. Coloque no sketch um *Calculate Map* e, dentro dele, dois functores *Number Map*.

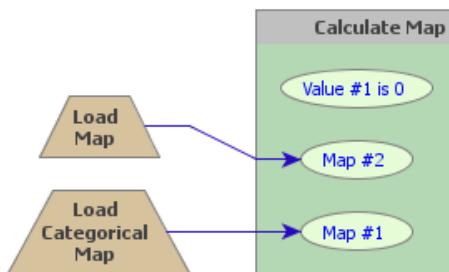


Agora atribua um número para cada *Number Map* e depois conecte *Load Map* e *Load Categorical Map* a eles.

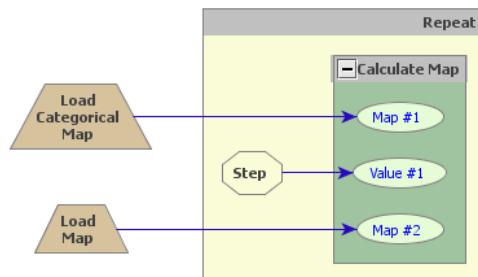


Carregue o mapa “amazon_states.tif” usando o functor *Load Categorical Map* e “Brazilian_amazon_landscape.tif” usando *Load Map*. Adicione ao *Calculate Map* um *Number Value* da aba Map Algebra Supplementary, atribua “1” a este e escreva a seguinte equação: **if i1 = v1 and i2 = 2 then 1 else null**. **DICA:** você pode copiar uma equação de um editor de texto e colá-la na caixa de equação.

Um valor é representado por **v# (v1, v2, ...vn)**. Feche o container. Não se esqueça de alterar **Data cell type** para “Unsigned 8 Bit Integer” e o valor nulo para “0”.



Pegue o container *Repeat* da aba Control e coloque-o no sketch. Arraste o *Calculate Map* para dentro dele. *Repeat* irá automaticamente ajustar seu tamanho para aninhar *Calculate Map*. Depois, adicione o functor *Step* dentro do *Repeat*. Abra *Calculate Map* clicando no ícone superior esquerdo de sua caixa e conecte *Step* ao porto **Value** do *Number Value*.



Abra *Repeat* com a ferramenta Edit Functor e insira “9”.



Isto implica que o modelo irá iterar 9 vezes. O functor *Step* se associa automaticamente ao container envolvente e passa para *Calculate Map* o valor da iteração corrente. Assim, para cada iteração, *Calculate Map* produz um mapa contendo a floresta remanescente (valor 2) para cada estado.

Agora você precisa somar todas as células que representam floresta. Observe que o resultado é um mapa binário apenas com valores um e nulo, este último representado por zero. Em seguida, use *Extract Map Attribute* (aba Map Algebra) para extrair a soma de células não-nulas. Este functor produz como resultado uma tabela com os seguintes atributos do mapa:

Chave Descrição

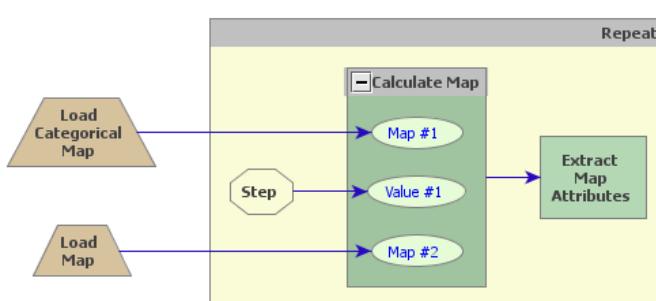
- 1 Número de linhas
- 2 Número de colunas
- 3 Número de células (número de linhas multiplicado pelo número de colunas).
- 4 Número de camadas
- 5 Altura da célula(em metros)
- 6 Largura da célula (em metros)
- 7 Área da célula (em hectares)

Atributos dinâmicos:

- 8 Número de células nulas
- 9 Número de células não-nulas
- 10 Valor mínimo (excluindo células nulas)
- 11 Valor máximo (excluindo células nulas)
- 12 Soma dos valores (excluindo células nulas)

Atributos estatísticos:

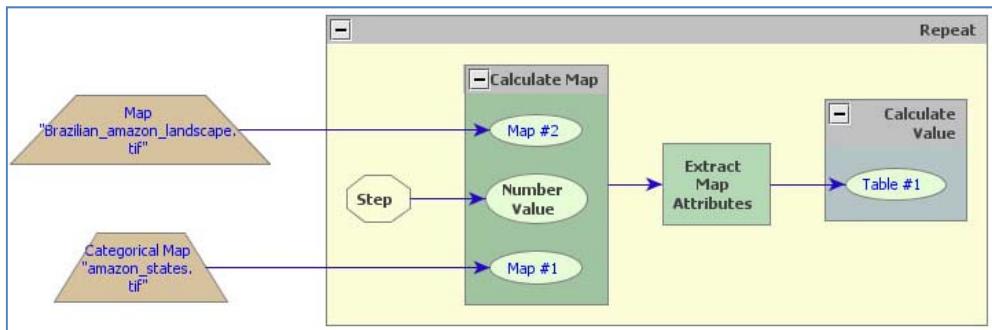
- 13 Média (excluindo células nulas)
- 14 Variância (excluindo células nulas)
- 15 Desvio padrão (excluindo células nulas)



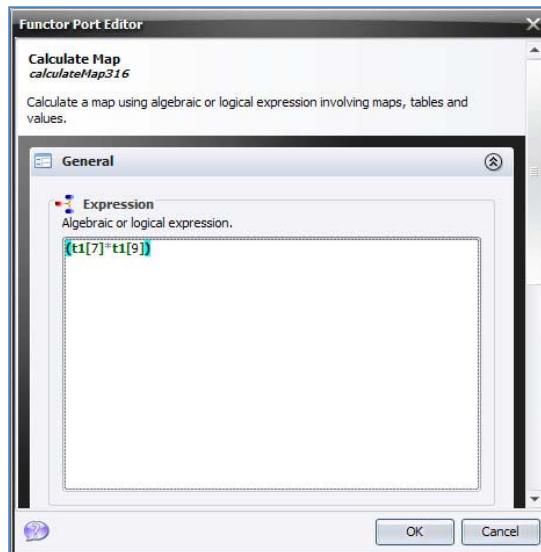
A extensão total em área é a soma das células não-nulas (campo 9) multiplicada pela área da célula em hectares (campo 7). Você precisará adicionar um *Calculate Value*, localizado na aba Table, para realizar este cálculo. Dentro dele, insira um *Number Table* (da aba Map Algebra Supplementary), o qual recebe a tabela de atributos resultado do *Extract Map Attributes*. Você precisará inserir “1” como identificador para esta tabela. Por fim, escreva: **t1[7]*t1[9]**

(área da célula (em hectares) * soma dos valores (excluindo células nulas))

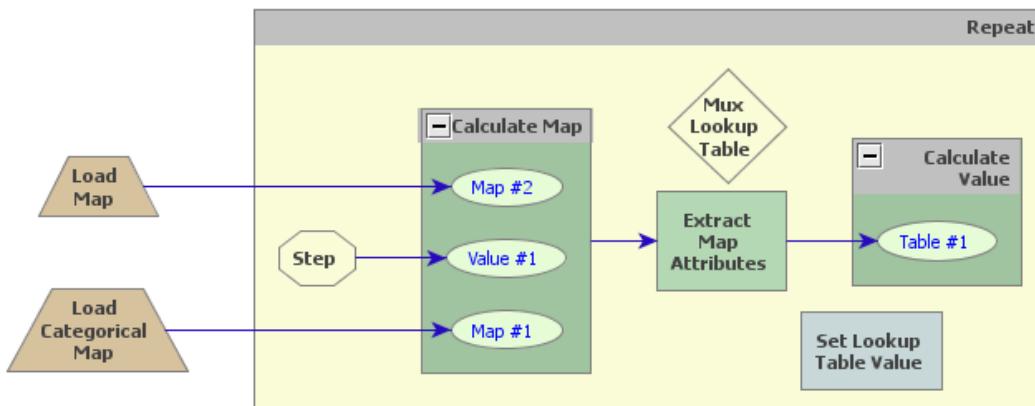
Observe que **t1** significa #1. Neste caso, os colchetes são necessários para representar o valor associado a uma chave da tabela.



Agora você precisa preencher a tabela para que ela registre a área calculada para cada estado. O functor *Set Lookup Table Value* atualiza a tabela inserindo um valor em uma posição definida por uma chave. Para preencher a tabela inteira, você precisa desenvolver um loop (ciclo fechado) que permite que este functor faça uma varredura em toda a tabela. Para fechar este loop, você precisará de um functor que é fundamental para o desenvolvimento de modelos dinâmicos. Assim, aqui introduzimos o conceito do functor *Mux*.



Um functor *Mux* pode ser um mapa, um mapa categórico, uma tabela ou um valor. Localize a aba Control para encontrar *Mux Lookup Table* e arraste-o para dentro do *Repeat*. Também arraste um *Set Lookup Table Value* a partir da aba Table.

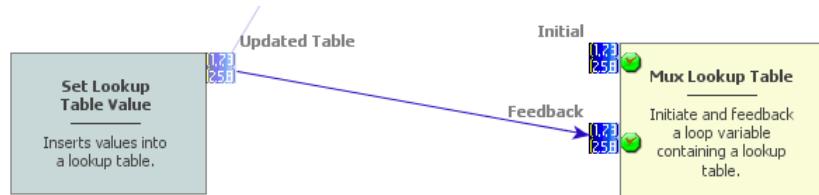


Agora, clique no *Mux Lookup Table* com o Edit Functor Ports. Cada functor *Mux* tem dois portos de entrada. Na primeira iteração, ele lê o arquivo de entrada do porto **Initial**; em seguida, ele recebe os dados da rodada do modelo através do porto **Feedback**. Este processo permite que os dados sejam atualizados pelo modelo, tornando-o dinâmico. Por isso, este functor é fundamental para incorporar retroalimentação em um modelo dinâmico. Abra também *Set Lookup Table Value* com o Edit Functor Ports.

Este functor recebe uma tabela que será atualizada com um valor inserido em uma posição definida por uma chave. Assim, você precisa conectar a tabela de saída do functor *Mux Lookup table* ao porto de entrada do *Set Lookup Table Value*.

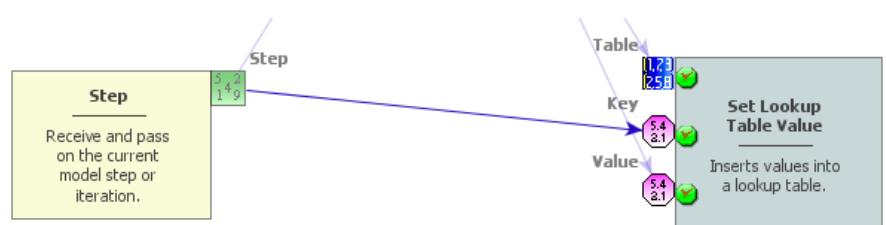
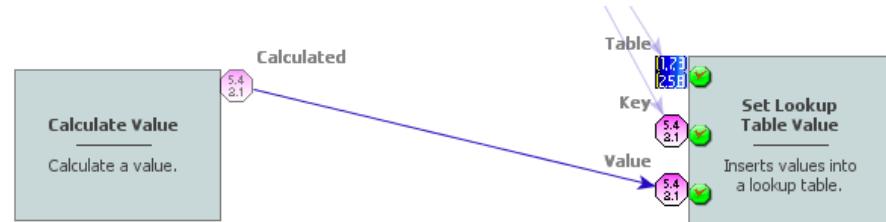
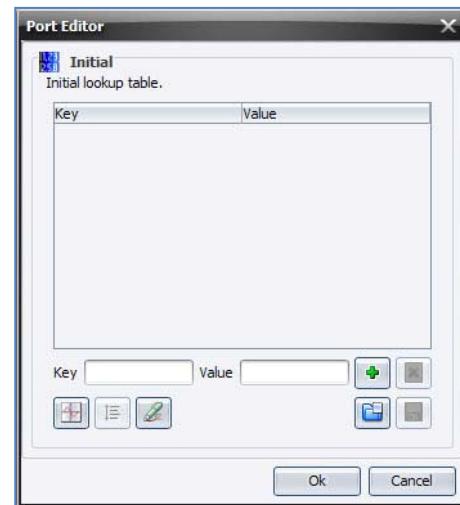


Novamente, vamos conectar a saída do *Set Lookup Table Value* ao *Mux Lookup Table*. Quando uma conexão tem duas ou mais opções, a janela *Edit Functor Ports* abre automaticamente. Você deve escolher o porto **Feedback**.

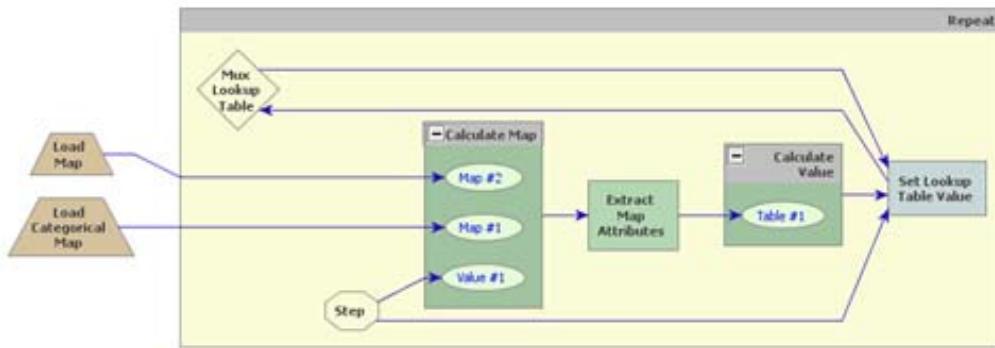


Agora clique no porto **Initial** com o botão direito do mouse. Você irá abrir um editor de tabela. Neste caso, você só precisa inserir “0, 0” como **Key** e **Value** para o primeiro registro da tabela, e então salvar estas entradas usando o botão ‘+’.

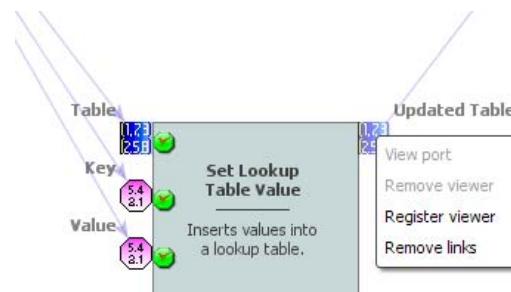
Além disso, você precisa conectar a saída do *Calculate Value* ao porto de entrada do *Set Lookup Table Value*. Neste momento, a janela *Edit Functor Ports* aparece porque há duas opções. Conecte a seta ao porto **Value**; a chave vem do passo corrente do modelo via conexão do *Step* ao *Set Lookup Table Value*.



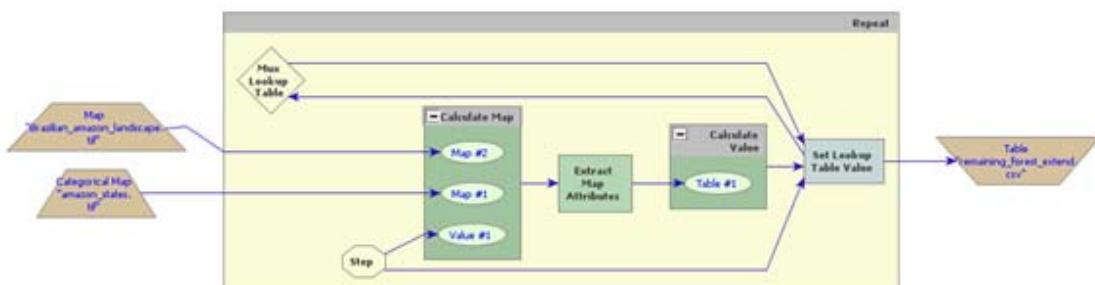
Clique na ferramenta de layout Left to Right e seu modelo irá se parecer como o que está logo abaixo. Observe que a conexão de retroalimentação é entre o *Mux Lookup Table* e *Set Lookup Table Value*. Como um último passo, você precisa salvar a tabela em um arquivo. Arraste o functor *Save Lookup Table* da aba Input/Output. Conecte *Set Lookup Table* a ele e edite o nome que será dado ao arquivo CSV. Embora **Suffix Digits** seja “2” por padrão, o nome do arquivo não terá um sufixo porque ele será salvo apenas ao final da execução do *Repeat*. **DICA:** Se *Save Map* for inserido dentro de um *Repeat*, um arquivo será salvo a cada passo do *Repeat* e um dígito representando o passo de tempo será adicionado ao final do nome do arquivo.



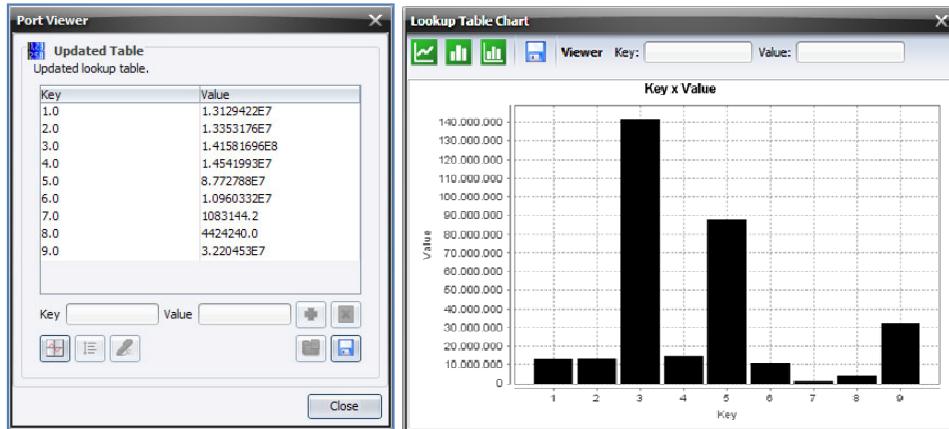
Por último, abra *Set Lookup Table Value* com o Edit Functor Ports. Clique em **Updated Table** com o botão direito e habilite **Register viewer**.



Teste a integridade do modelo, salve-o e, se tudo estiver O.K., clique no botão de execução. Isto poderá levar certo tempo.



Vá para *Set Lookup Table Value*, abra-o com *Edit Functor Ports* e clique com o botão direito sobre **Updated Table** para ver o resultado. Faça também um gráfico, clicando no botão de gráficos na parte inferior esquerda.



Estas são as extensões em hectares da floresta Amazônica remanescente por estado. **DICA:** Você pode também abrir o arquivo CSV com um programa de planilhas.

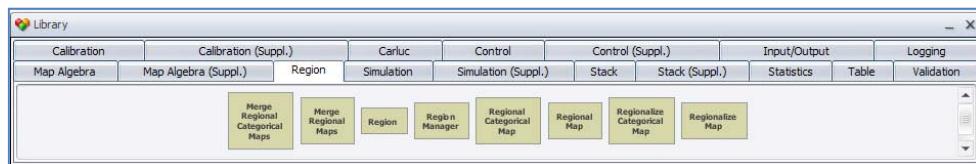
Você completou com sucesso as duas lições introdutórias. Outro modo de resolver o problema da lição 2 é através do uso de um subconjunto de functores Region. Este será o assunto da próxima lição.

4. Lição 3: Usando o conceito de região

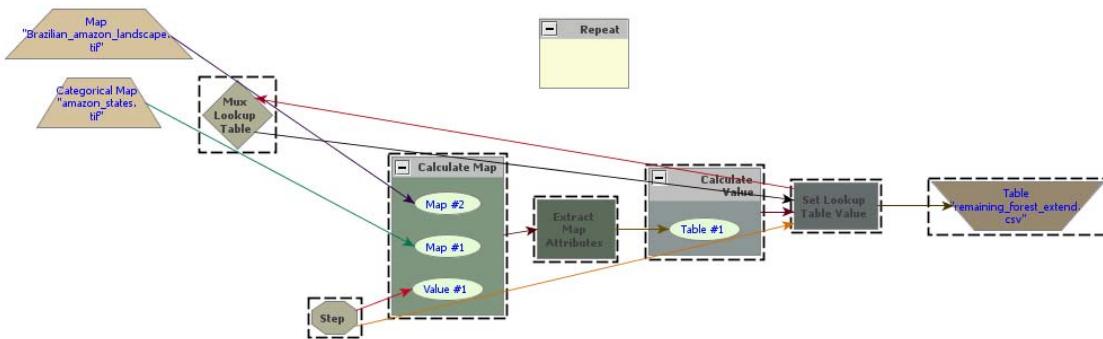
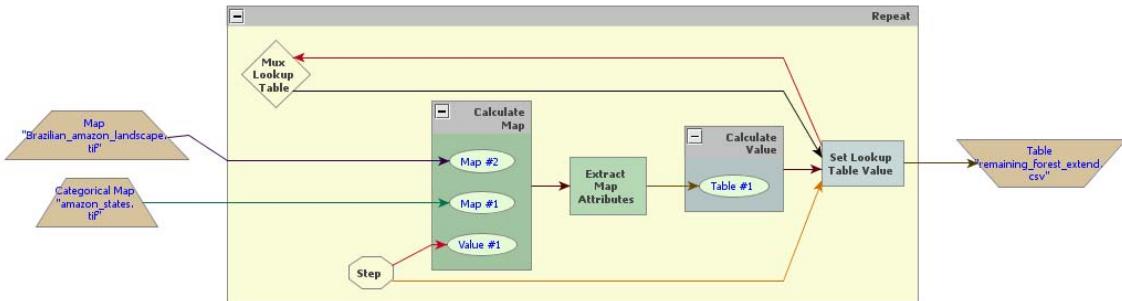
O que você irá aprender?

- Como usar o conceito de região
- Functores:
 - *Region manager*
 - *For Each Category*

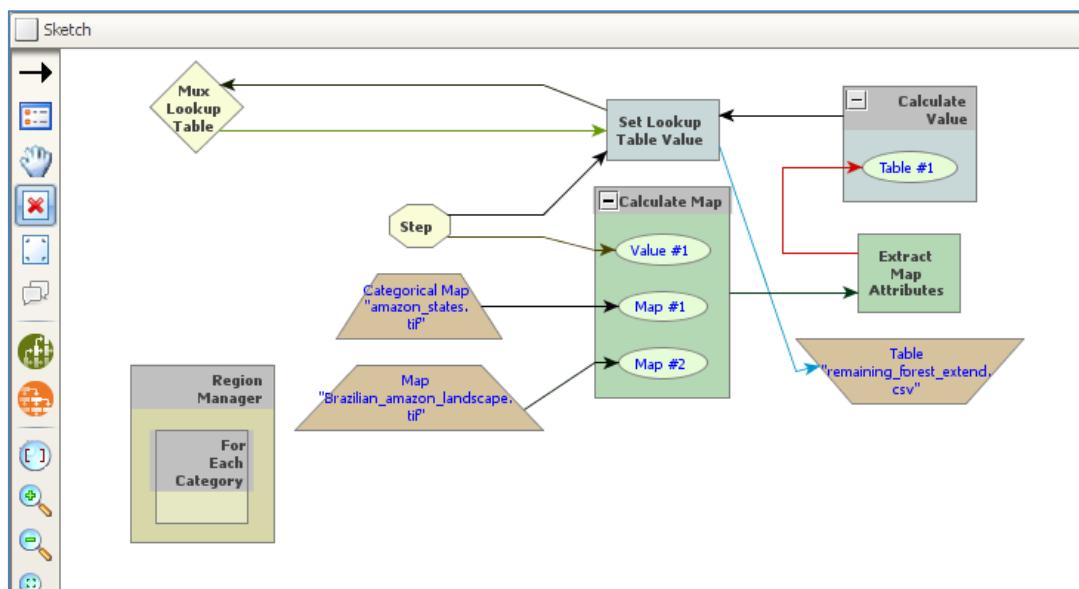
Um modo alternativo de resolver o exercício da **Lição 2** é através de Region. Este subconjunto de functores permite a divisão de um mapa em vários mapas regionais, os quais podem se tornar uma entrada para modelos separados ou submodelos, cujos parâmetros são personalizados para cada região do mapa. Vamos adaptar o modelo anterior para utilizar o conceito de regiões. Primeiro, abra a aba Region na janela library.



Estes functores podem ser combinados para produzir mapas de regiões e submodelos, assim como para unir mapas de regiões em um único mapa depois que um processamento é executado. Nesta lição você irá aprender como obter a área de floresta remanescente para cada estado; como resultado, cada mapa regional irá corresponder a uma área estadual. Abra o modelo completado na pasta **lesson 2**. Selecione os functores dentro do container *Repeat* com a ferramenta Hand e arraste-os para fora do *Repeat*. **DICA:** Use o **Crtl + botão direito** para selecionar mais de um functor, isso também evitaria que você selecione *Repeat* sem querer. O functor *Repeat* irá encolher.



Apague o container *Repeat* e insira os containers *Region Manager* da aba *Region* e *For Each Category* da aba *Control*. Coloque o último dentro do primeiro.

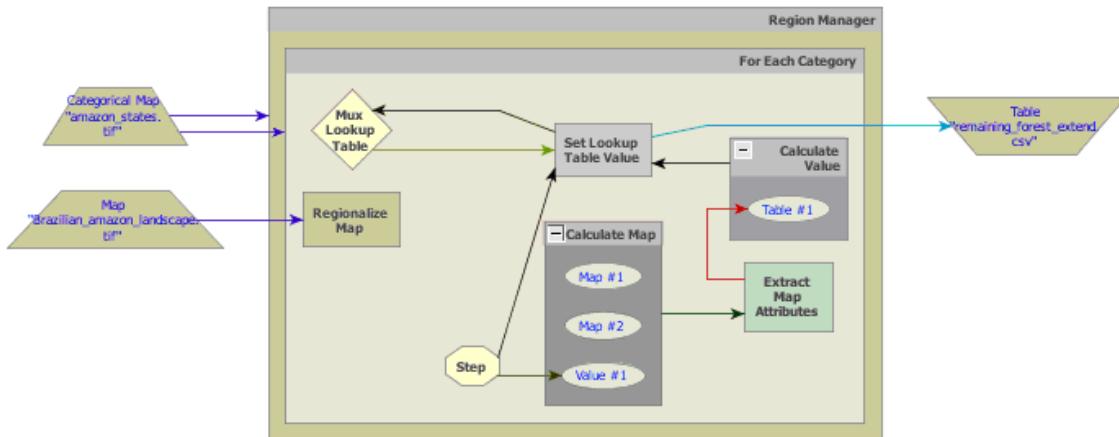


Region Manager conduz o processo de regionalização, enquanto *For Each Category* substitui *Repeat* fazendo com que o modelo itere para cada categoria presente no mapa categórico usado para definir as regiões, neste caso, o mapa dos estados amazônicos brasileiros.

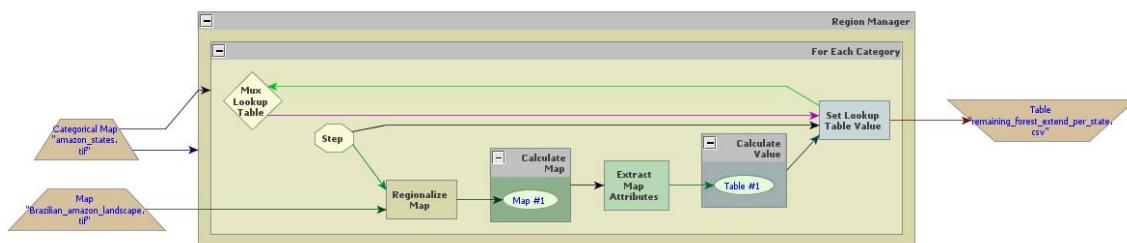
TIP: Enquanto *Repeat* itera sequencialmente, *For Each Category* itera de acordo com as categorias do mapa, o que não necessariamente é sequencial.

Agora selecione todos os functores, exceto os mapas de entrada e saída, e coloque-os dentro de *For Each Category*. **DICA:** é mais fácil selecionar tudo a ser inserido dentro do *For Each Category* e depois retirar apenas *Load Map* e *Save Map*.

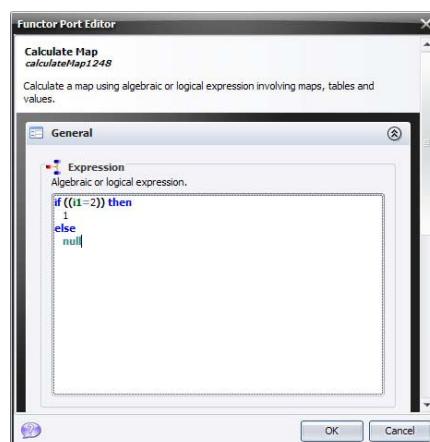
Agora quebre a conexão entre o *Categorical Map* e *Calculate Map* (use a ferramenta Remove Items e clique na seta que conecta os dois functores); em seguida, conecte o primeiro ao *Region Manager* e também ao *For Each Category*. Este mapa irá controlar o processo de regionalização dividindo os outros mapas em várias regiões de acordo com suas classes de células ou categorias através do functor *Regionalize Map*. Arraste este functor e insira-o dentro do *For Each Category*, quebre a conexão entre Map “brazilian_amazon_landscape.tif” e *Calculate Map* e, por fim, conecte o primeiro a *Regionalize Map*.



Observe que há dois functores *Number Map* desconectados dentro do *Calculate Map*. Você não irá precisar mais do *Map # 2* e do *Value # 1*, portanto, apague-os. Agora conecte o *Regionalize Map* ao *Map # 1*. Abra o functor *Table* e mude o nome do arquivo para “*remaining_forest_extent_per_state*”. Certifique-se de que ele será salvo dentro da pasta **lesson3**. Você precisará conectar *Step* ao *Regionalize Map* permitindo que ele passe ao primeiro a identificação da região atual proveniente de *For Each Category*.



Por último, você precisará modificar o *Calculate Map*. Abra-o com o Edit Functor. Escreva: ***if i1 = 2 then 1 else null***. Lembre-se de que “2” representa floresta.



Por fim, salve o modelo como um novo arquivo “my_calc_forest_remaining_per_state_using_subregions.xml” dentro da pasta **lesson3**.

Verifique a integridade do modelo e, se tudo estiver O.K., clique no botão de execução. Compare o resultado deste modelo com o resultado do modelo da lição anterior. Os resultados são equivalentes? Você é capaz de explicar como este modelo funciona? Observe que você não precisou separar a informação por estado no *Calculate Map* porque *Regionalize Map* já tinha feito isso. **DICA:** a abordagem de regiões é uma forma útil de se quebrar o mapa em vários subconjuntos seja para personalizar um submodelo, por exemplo, para rodar com diferentes parâmetros por país, estado ou município, ou para otimizar a memória em uso, uma vez que o modelo não precisa lidar com todas as células do mapa para um cálculo específico ou processamento, mas apenas as células selecionadas de uma região por vez. Agora, vamos avançar em direção a uma análise espacial mais avançada.

5. Lição 4: Calculando uma superfície de custo acumulado e um caminho de menor custo

O que você irá aprender?

- Como calcular uma superfície de fricção, superfície de custo e um caminho de menor custo
- Functores:
 - *Calc Cost map*
 - *Calc Pathway map*

Este exercício requer o cálculo de uma superfície de fricção, representando o custo relativo de atravessar uma unidade de célula dependendo do uso da terra. Podemos expressar esta superfície tanto em termos de distância – em que não há custo diferencial entre tipos de usos da terra; de forma que o caminho de menor custo será a rota mais curta, i.e. a distância Euclidiana – tempo, custo financeiro ou algum tipo de esforço. Assim, este valor é calculado em relação a alguma unidade (tempo, custo de transporte, etc).

Neste exercício, nós exploraremos o uso dos functores *Calc Cost Map* e *Calc Pathway Map*. Para encontrar a solução ótima para a superfície de custo acumulado, o functor *Calc Cost Map* usa um algoritmo heurístico que varre repetidas vezes um mapa até que a melhor superfície de custo é obtida. Com o aumento do número de passos, também aumenta a proximidade de uma solução ótima. Use “0” como número máximo de passos para uma solução ótima, mas, mas em geral, apenas dois passos são suficientes para obter uma superfície bem próxima da solução ótima.

Questão:

Queremos definir o caminho de menor custo para uma estrada de ferro que irá conectar uma ferrovia a uma cidade localizada na região. Do ponto de vista dos engenheiros, o caminho deverá ser o mais barato, i.e. o mais curto, mas alguns tipos de uso da terra não podem ser convertidos em espaços abertos para estradas, representando assim barreiras, e outros tem um custo muito elevado para atravessar. Nossa tarefa é determinar o caminho de menor custo entre a cidade e a ferrovia existente.

O conjunto de dados empregados neste exercício compreende:

1. Um mapa de uso e cobertura da terra da região Nordeste do Mato Grosso, Amazônia Brasileira (fig. 7). (*landuse.tif*)
2. Um mapa de declividade. (*slope.tif*)

3. Um mapa com a cidade a ser conectada. (town1.tif)
4. Um mapa com a atual ferrovia (railroad.tif)

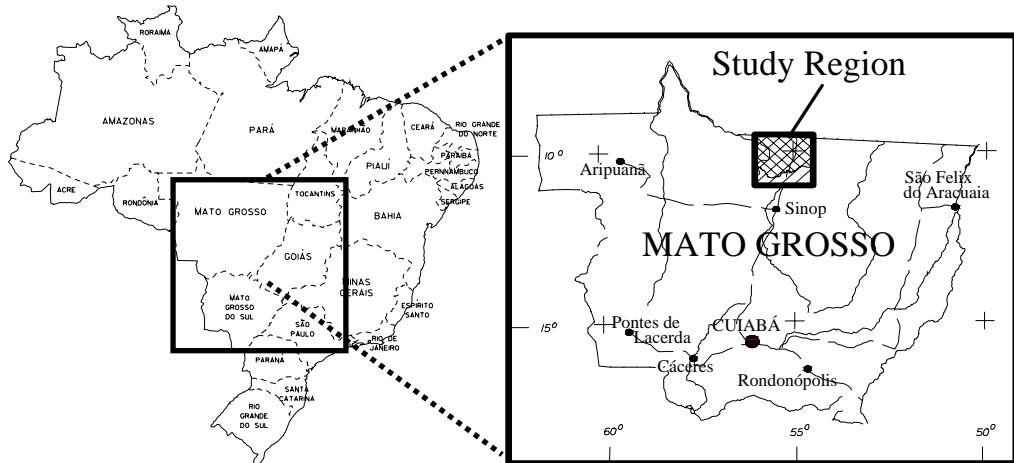


Fig. 7 – Nordeste do Mato Grosso e sua localização no Brasil.

Abra Dinamica EGO e carregue os mapas mencionados acima no Map Viewer a partir da pasta **lesson4\originals** usando a paleta de cores “mt”. Abra o mapa de declividade usando “Pseudocolor” como a paleta de cores e, no Histograma clique em Limits to Actual e em Histogram Equalize. De início, você precisará reclassificar o mapa de uso da terra para representar o custo de atravessar cada um dos tipos de uso da terra. Você também precisará reclassificar o mapa de declividade e então combiná-lo com o mapa de uso da terra. Para o mapa de uso da terra, use a seguinte tabela:

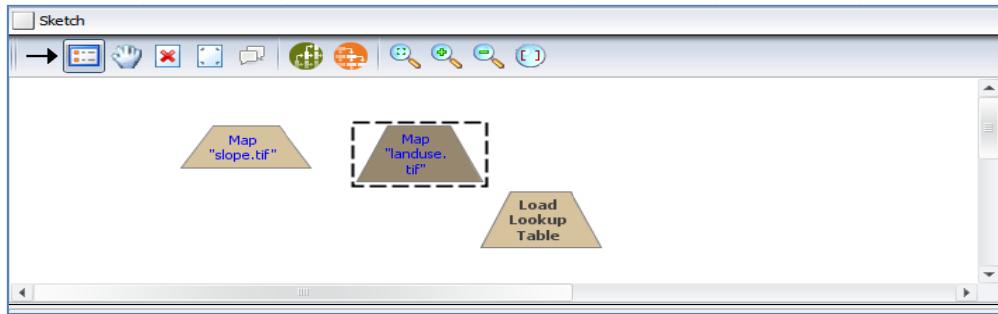
Uso da Terra	Atributo	Fricção	Explicação
Planícies de inundação	0	10000	Praticamente uma barreira
Pequenos Rios	1	50	É necessário construir pontes
Pastagem	2	1	Custo básico
Regeneração	3	10	É necessário cortar arbustos e pequenas árvores
Remanescente de Floresta	4	500	É necessário obter licença para suprimir a vegetação nativa, a qual possui um valor de preservação intrínseco
Meio Urbano	5	10000	Praticamente uma barreira
Estradas	6	30	É necessário construir viadutos ou instalar semáforos para controlar o tráfego

A fricção cresce em função dos intervalos de declividade, como se segue:

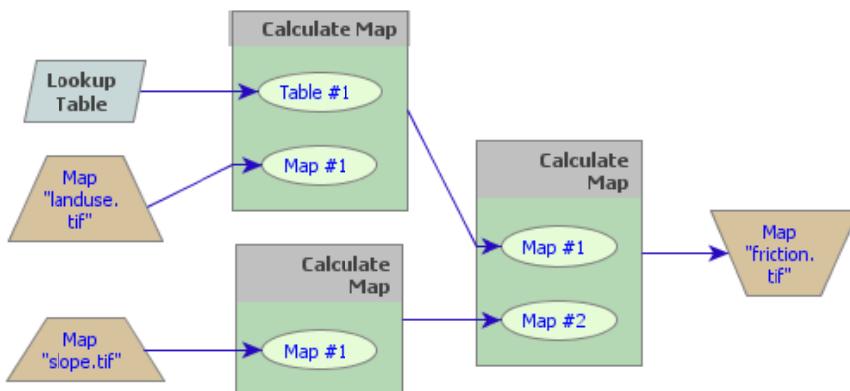
Declividade (graus)	Fricção
0-1	1
2-5	1.3
5-10	1.5
10-15	1.9
15-20	2.5
> 20	5

Vamos começar o modelo carregando os mapas “landuse.tif” e “slope.tif”, usando o functor *Load Map*. Em seguida, vamos incorporar as duas tabelas anteriores. Adicione um *Lookup Table* da aba *Table*.

Você deverá ter algo como o seguinte:

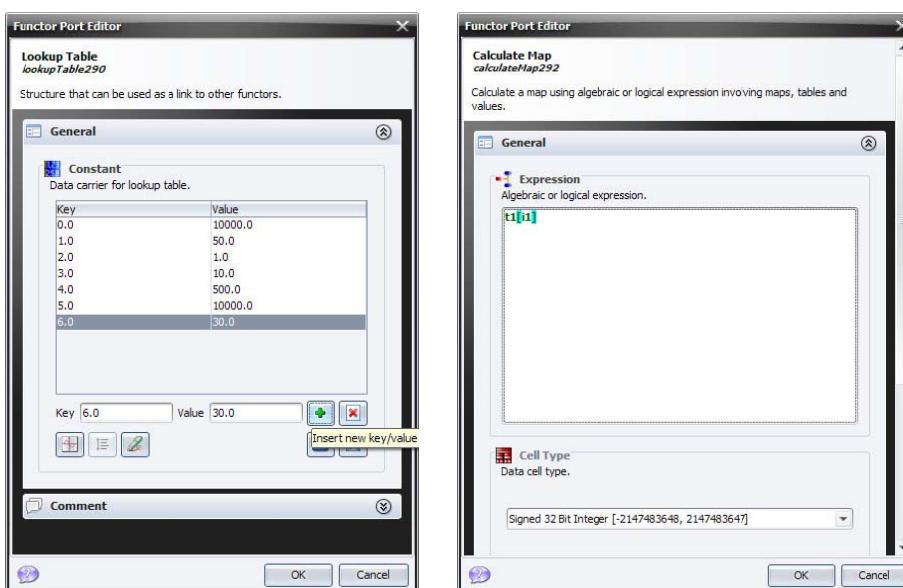


Agora insira três *Calculate Map* e quatro *Number Map* functores, um dentro de cada um dos dois primeiros *Calculate Map* e dois *Number Map* functores dentro do terceiro, e um *Number Table* dentro de um dos dois primeiros functores *Calculate Map*, e um *Save Map*. Abra os functores *Number Map*, insira um número único para cada um (1 e 2) e insira “1” ao *Number Table*. Por fim, conecte *Map “landuse.tif”* ao *Number Map 1* e *Map “slope.tif”* ao *Number Map 2* dos dois primeiros *Calculate Map*. Em seguida, conecte os dois primeiros ao terceiro *Calculate Map* e este ao *Save Map*. Abra o *Save Map* e dê o nome “friction.tif”. Mude a pasta para um nível acima, depois mude o formato para “geotiff”. Isto é o que você terá:



Agora, você precisa inserir a tabela de uso da terra no functor *Lookup Table*. Abra-o com o *Edit Functor* e comece adicionando chaves e valores (preencha os campos e pressione o botão de adição):

Abra o *Calculate Map* que contém o *Number Table* e escreva a seguinte fórmula: **t1[i1]**



Esta fórmula irá obter o valor do mapa e usá-lo como chave para acessar a tabela, classificando, assim, o mapa de acordo com os valores de custo.

Insira a seguinte equação dentro do segundo *Calculate Map*:

if i1 < 1 then 1 else if i1 < 5 then 1.3 else if i1 < 10 then 1.5 else if i1 < 15 then 1.9 else if i1 < 20 then 2.5 else 5

Esta expressão corresponde à tabela de fricção por declividade.

No terceiro *Calculate Map* insira:

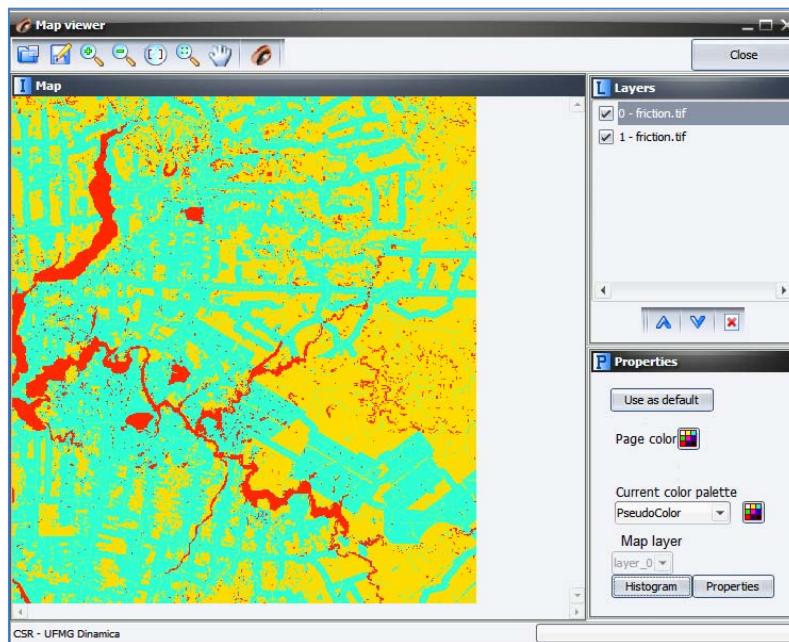
i1*i2

Salve o modelo como “my_friction”, verifique sua integridade e, se tudo estiver O.K., execute-o.

Abra o arquivo “friction.tif” no Map Viewer usando o “Pseudocolor” como **Current Color Palette** e no Histograma, clique em **Limits to Actual** e **Histogram Equalize**.

O que você vê? Observe que a cor vermelha representa as áreas com alto custo de travessia.

DICA: Sinta-se livre para usar outro software para visualizar os mapas. Geotiff é automaticamente aberto nos pacotes do ARC GIS, ER MAPPER ou ERDAS.



Vamos para a segunda parte desse exercício. Carregue “town1.tif” da pasta \originals usando *Load Map* e “railroad.tif” usando *Load Categorical Map*. Lembre-se que este functor categoriza o mapa.

Arraste um *Calc Cost Map* e um *Calc Pathway Map* da aba Map Algebra e também um *Save Map*.

Algumas observações a respeito deste algoritmo:

O algoritmo que calcula o mapa de custo é um tipo geral de "Pushbroom". Entretanto sua performance espacial se aproxima do conhecido algoritmo "Pushgrow", especialmente quando é usado dois ou mais passos.

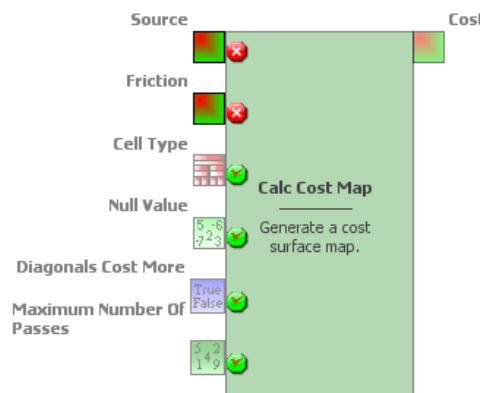
Por padrão, as dimensões das células (largura e altura) não são consideradas no cálculo de custo. Para levar isso em consideração, habilite a opção **Friction is relative** em Advanced Options.

Penalização por movimentos diagonais é efetivo apenas quando os valores de fricção são altos ou o mapa de custo é representado usando números reais.

Mapas de fricção com células representadas em números reais requerem mapas de custo com células representadas também em números reais. Caso contrário um erro será reportado.

Cada passo usado para calcular o mapa de custo corresponde a quatro passadas no mapa originadas a partir de direções opostas.

Lugares inalcançáveis no mapa de fricção são excluídos no mapa de custo e, assim, são representados como células com valores nulos. O custo não é acumulado através das células de valores nulos, assim, regiões rodeadas por células de valores nulos não terão seus custos calculados, a menos que haja uma feição dentro desta região.

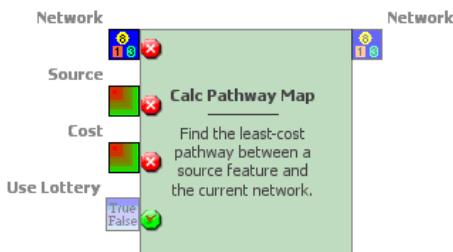


O porto **Source** receberá o mapa "railroad.tif" e o mapa de fricção resultante do terceiro *Calculate Map*. Habilite a opção **Diagonals Cost More**. Isto irá penalizar o movimento através de células diagonais. Ajuste o **Maximum Number of Passes** para "2". Deixe as demais opções sem alteração.

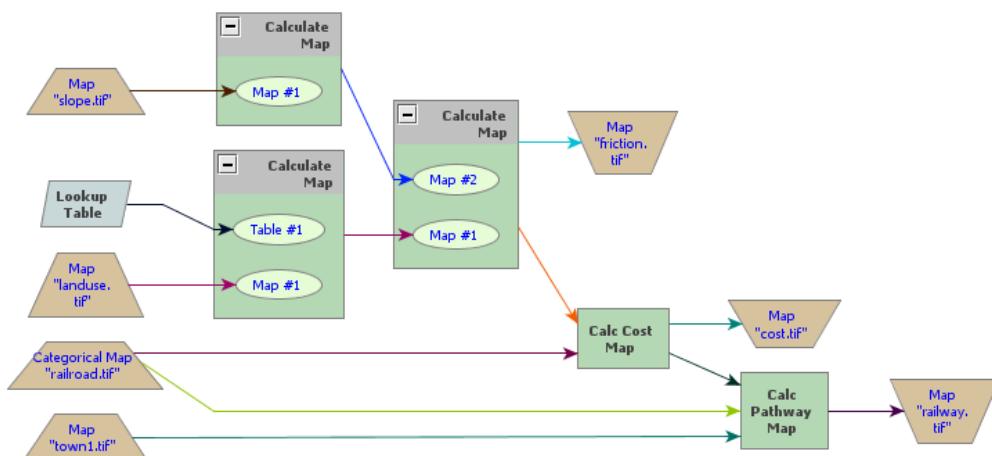


Agora abra o *Calc Pathway Map* com o *Edit Functor Ports*. Conecte o mapa "town1.tif" ao porto **Source** (**DICA**: Source, neste caso, também representa o destino, uma vez que o mapa de custo foi construído a partir da estrada existente. Assim, este algoritmo irá procurar pelo caminho de menor custo a partir da fonte até a feição já existente, i.e. a estrada de ferro), conecte o mapa de saída do *Calc Cost Map* ao porto **Cost** e *Map railroad.tif* ao porto **Network** (porque este mapa representa uma rede de feições lineares) e o porto **Network** de saída ao *Save Map*. Ative a opção **Use Lottery** (este é um artefato que permite que o modelo escolha o caminho quando dois ou mais mínimos locais são encontrados).

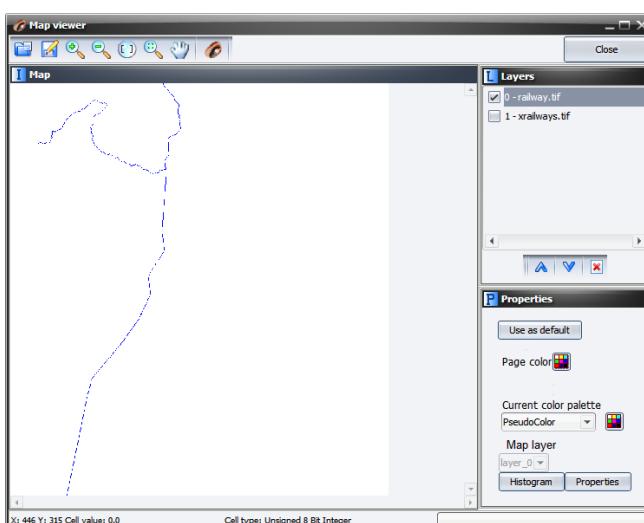
Calc Pathway Map ignora células com valores iguais ou inferiores a 0 ou células nulas. Por sua vez, *Calc Cost Map* necessita de um mapa de rede com células nulas representando locais sem feições. Vá para *Categorical Map* e abra-o usando Edit Functor. Habilite a caixa **Define Null Value** e certifique-se de que **Null Value** está representado pelo valor “0”.



Clique em *Save Map* com o *Edit Functor*, mude a pasta para um nível acima, mude o formato do arquivo para “Geotiff”, insira “0” para o **Suffix to Digits** e, por fim, dê o nome “railway.tif”. O modelo final irá se parecer com o seguinte:

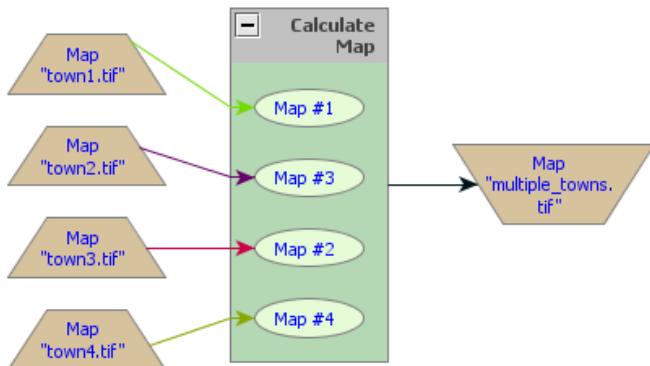


Salve o modelo em um novo arquivo, nomeando-o “my_pathway.xml”, verifique e se tudo está certo, execute-o. Isto vai demorar apenas um pouquinho. Dinamica EGO possui desempenho superior em relação à maioria dos pacotes comerciais de SIGs; tente rodar este modelo em outro software apenas para comparação de performance. Abra o arquivo “railway.tif” no Map Viewer usando “PseudoColor” como **Current Color Palette**. O que você vê?



Você pode também tentar maximizar a solução para o algoritmo *Calc Cost Map* ajustando o **Maximum Number of Passes** para “0”. Compare o tempo gasto por esta execução e o caminho resultante dos do modelo anterior. Houve alguma grande diferença?

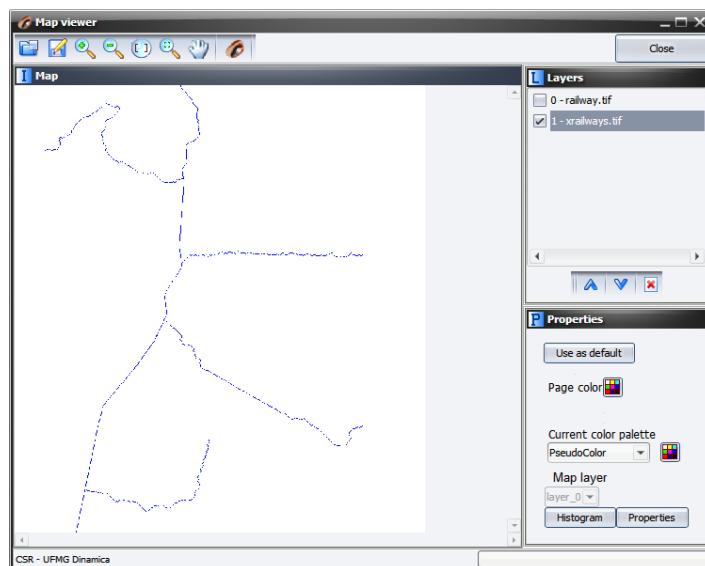
Este tipo de modelo também pode ser modificado para desenvolver múltiplos caminhos simultaneamente. Abra o modelo “join_towns.xml” na pasta **lesson 4**.



Este modelo mostra como você pode usar *Calculate Map* para combinar informações de diferentes mapas em um único mapa. O resultado será um mapa representando as células centrais para quatro cidades. **DICA:** utilize sempre uma única célula para representar uma localização a ser alcançada pelo *Calc Pathway Map*.

Agora substitua a entrada em *Map “town1.tif”* pelo arquivo “multiple_towns.tif” e mude o arquivo em *Map “railway.tif”* para “xrailways.tif”.

Você obteve algo como o mapa abaixo?



Se você for até Examples\run_lucc_northern_mato_grosso\run_roads_with_comments e abrir o modelo “mato_grosso_road.xml”, você verá como este conjunto de algoritmos pode ser adaptado e combinado para construir um Road Constructor Module, um submodelo que simula a expansão da malha rodoviária na região da fronteira amazônica. Este modelo é um exemplo da habilidade da plataforma Dinamica EGO para a criativa construção de modelos espaciais.

6. Lição 5: Avaliação Multicritério no Dinamica EGO

O que você irá aprender?

- AMC para planejamento urbano e regional.
- Functores:
 - *Calculate Distance to Feature Map*
 - *Group*

A Avaliação Multicritério (AMC) é um método frequentemente utilizado para avaliações de impactos ambientais e para planejamento urbano e regional. Neste exercício você irá aplicar o AMC para identificar áreas que são favoráveis para o desenvolvimento de uma nova cidade no Nordeste do Mato Grosso – uma região de fronteira amazônica –, mitigando, ao mesmo tempo, possíveis impactos ambientais decorrentes do assentamento urbano. Novamente, nós nos deparamos com dois pontos de vista, um dos desenvolvimentistas e outro dos conservacionistas. Um modo de resolver esta questão é listando todos os critérios que favorecem o estabelecimento de uma nova área urbana e os que o restringem ou o impedem.

Neste problema, nós temos dois tipos de critérios, aqueles que restringem nossa análise apenas para áreas específicas, sendo assim booleanos, com valor 1 ou 0, e aqueles que admitem um certo grau de adequabilidade para um local dependendo de seus atributos biofísicos e de infraestrutura.

Nossos critérios são:

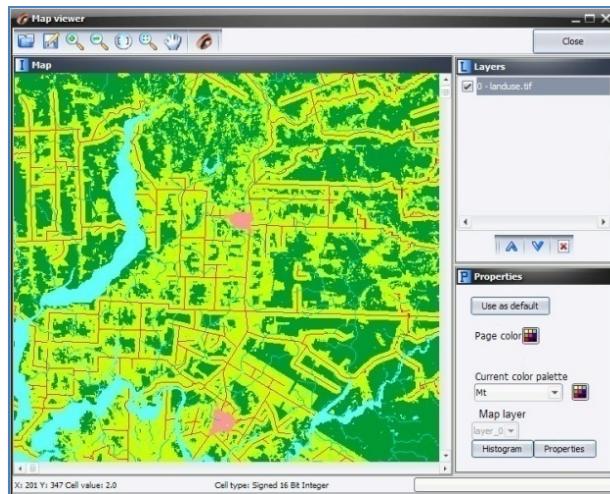
- 1) Distância até estradas principais < 15 quilômetros.
- 2) Tempo de viagem até cidades vizinhas < 30 minutos
- 3) Distância mínima de 10 km até cidades existentes.
- 4) Distância mínima de 1000 metros até áreas inundadas (Malária é endêmica nessas regiões)
- 5) Somado a isso, temos que considerar que nem todos os terrenos estão disponíveis. Nós não queremos encorajar mais desmatamento na região, portanto, precisamos usar apenas áreas já desmatadas ou abandonadas. Também devemos excluir áreas urbanas já existentes, rios e planícies de inundação.
- 6) Áreas no mínimo maiores que 1000 hectares.
- 7) Declividade média < 0.5 graus.

6.1 Primeiro Passo: Encontrando áreas sem impedimentos

O mapa de uso da terra tem as seguintes classes:

Uso da terra	id
Planícies de inundação	0
Pequenos rios	1
Pastagem	2
Vegetação em recuperação	3
Remanescente de floresta	4
Área urbana	5
Estradas	6

Carregue “\lesson4\originals\landuse.tif” no Map Viewer e mude o **Current Color Palette** para “mt”.

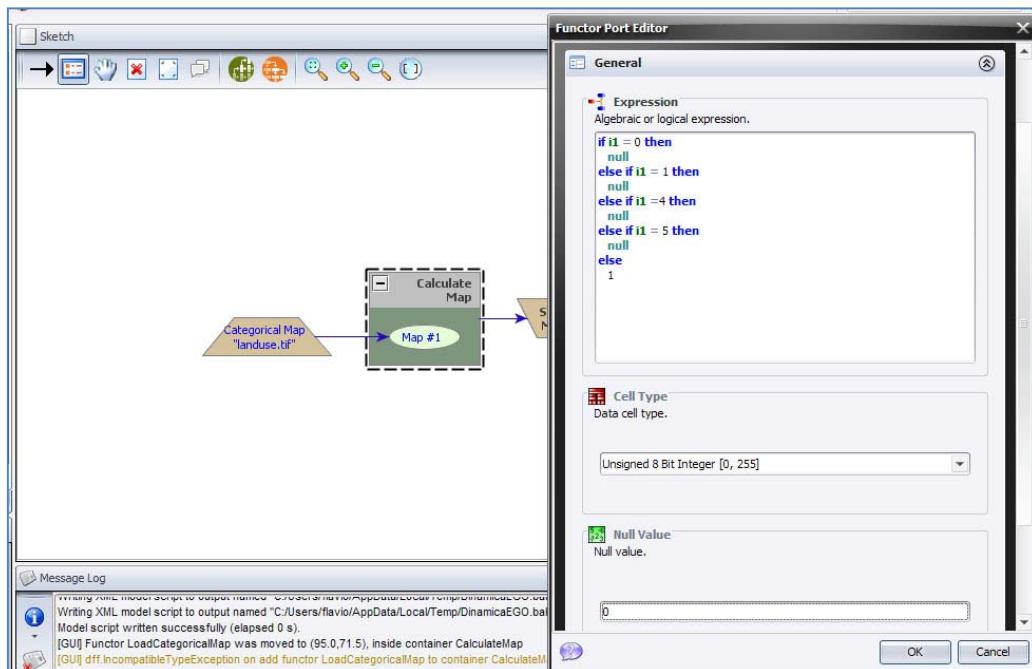


Comecemos montando o modelo. Insira um *Load Categorical Map* no plano de sketch e abra o arquivo “\lesson4\original\landuse.tif”. Agora insira um *Calculate Map* e, em seguida, um *Number Map* dentro dele. Escreva “1” no **Map Number** dentro de *Number Map*. Agora insira um *Save Map* e nomeie como “unconstrained_areas.tif”, certifique-se de que você mudou a pasta para **lesson5**. Altere o formato de arquivo para **Geotiff** e insira “0” em **Suffix Digits** (**DICA**: você pode melhorar o desempenho do modelo evitando salvar mapas intermediários). Nesta etapa, você irá salvar o mapa apenas para verificar o resultado desta operação. Não se esqueça de conectar todos os functores.

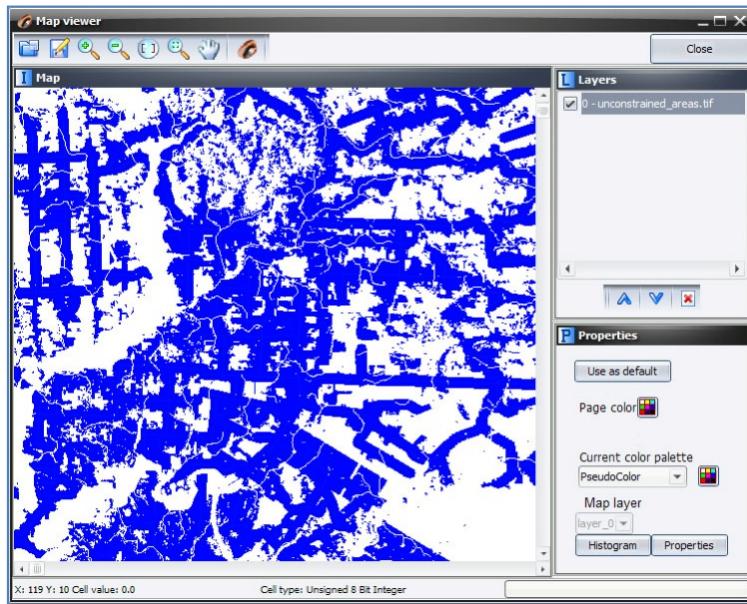
Agora abra o *Calculate Map* com o Edit Functor e escreva a seguinte equação:

if i1 = 0 then null else if i1 = 1 then null else if i1 = 4 then null else if i1 = 5 then null else 1

O valor nulo irá esconder rios, planícies de inundação, áreas urbanas e estradas em operações subsequentes. Não se esqueça de mudar **Data Cell Type** para “Unsigned 8 Bit Integer” e **Null Value** para “0”.



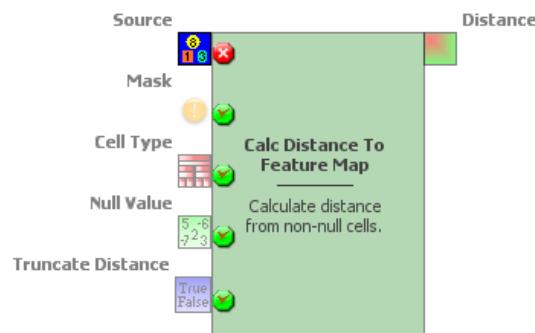
Salve o modelo como “My_MCE_part1.xml”, execute-o e examine o mapa resultante no Map Viewer.



6.2 Segundo Passo: Construindo buffers para definir áreas inapropriadas

Nesta etapa você precisará construir buffers (zonas de amortecimento) para considerar o critério de distância de cidades e áreas inundadas.

Carregue um *Calculate Categorical Map* junto com o seu *Number Map* (**DICA:** A única diferença entre o *Calculate Map* e o *Calculate Categorical Map* é que o último gera como resultado um mapa categórico). Você irá precisar de um *Calculate Categorical Map* para usar o functor *Calc Distance to Feature Map*; você também pode acessá-lo através da aba Map Algebra. Embora não seja necessário, você pode salvar o mapa utilizando um *Save Map*. Salve-o como “distance_to_towns.tif”. Agora conecte o mapa categórico “landuse.tif” ao *Number Map*, em seguida, ajuste o **Map Number** para “1”. Abra o *Calc Distance to Feature Map* com o Edit Functor Ports.



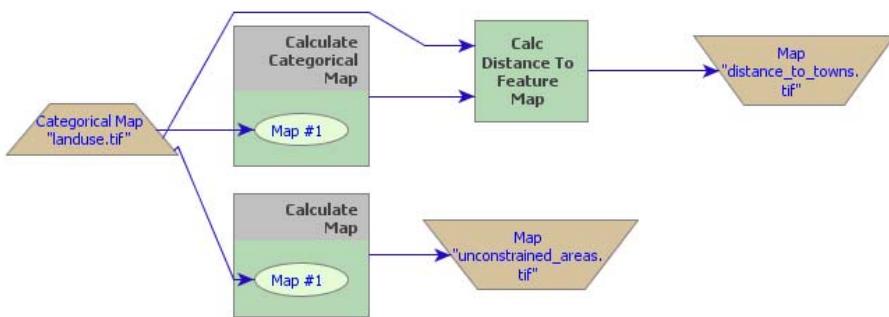
O porto **Source** é um mapa contendo as feições para as quais as distâncias serão calculadas. Uma feição é representada por um valor não nulo. O porto **Mask** é um mapa usado para mascarar o cálculo de distância nas células nulas. Vamos conectar a saída do *Calculate Categorical Map* em **Source** e o mapa categórico “landuse.tif” em **Mask**.

Agora abra *Calculate Categorical Map* e escreva a seguinte equação:

if i1 = 5 then1 else null

Lembre-se de que 5 é o número de identificação para a classe área urbana. Você não precisa mudar **Data cell type** nem o **Null Value** nos dois functores, embora, para economizar memória, você poderá mudar **Data cell type** para “Signed 8 Bit Integer” e **Null Value** para “0” no *Calculate Categorical Map*. Você pode também habilitar a opção **Truncate Distance** em *Calc Distance to Feature Map*. Isto irá evitar que se extrapole a representação numérica máxima para **Data cell type**.

Agora conecte o porto **Distance** em *Calc Distance to Feature Map* ao *Save Map*. O modelo deverá estar pronto para rodar. Sempre salve o modelo antes de colocá-lo para executar. Salve-o como “my_MCE_part1&2.xml”. Examine o resultado com Map Viewer.

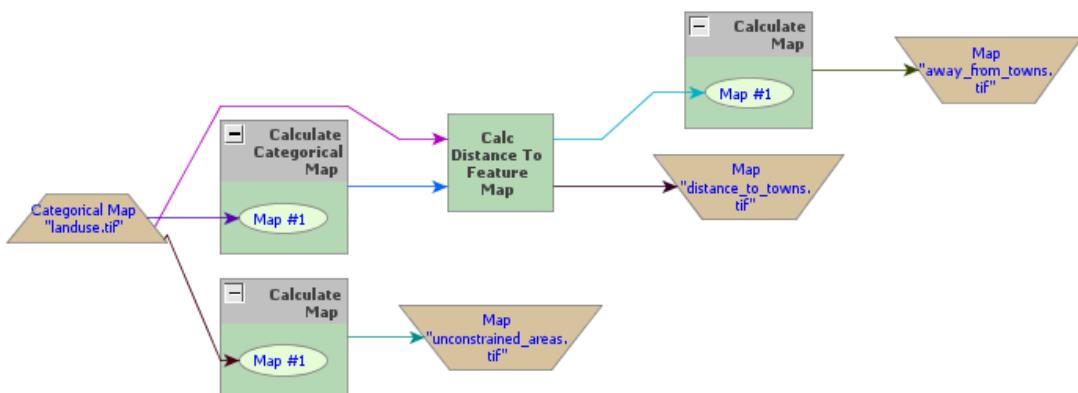


Como um último procedimento, adicione um *Calculate Map* e seu *Number Map*, conecte a saída de *Calc Distance to Feature Map* ao seu *Number Map* e salve o resultado usando um *Save Map* com o nome “away_from_towns.tif”. Observe que neste estágio, nós estamos pulando alguns passos, supondo que eles já foram bem compreendidos até agora. Escreva o seguinte na caixa de equação do *Calculate Map*:

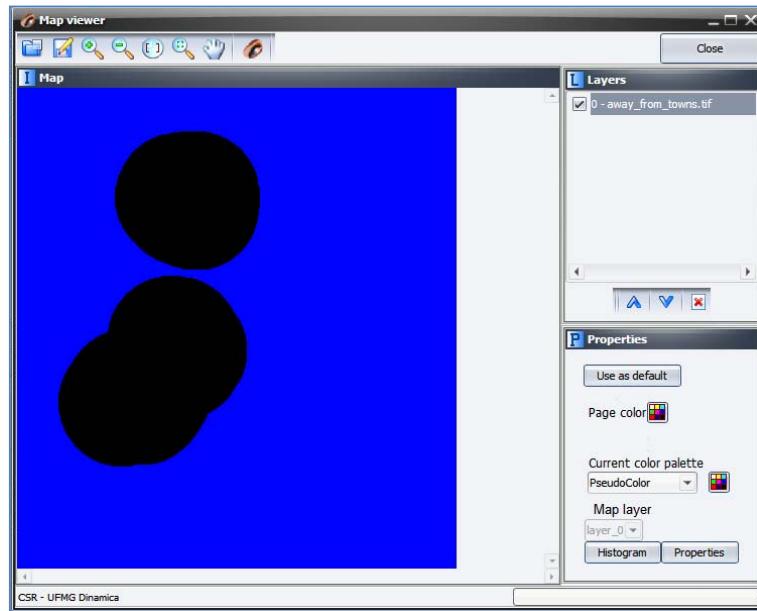
if i1 < 10000 then null else 1

DICA: O mapa de distância é sempre produzido em metros; o critério de restrição é sempre ajustado para nulo.

Você obterá o seguinte modelo:



Salve-o e execute-o. Abra o mapa “away_from_towns.tif”. É isto o que você obteve?



Como você irá solucionar o problema tendo os seguintes critérios?

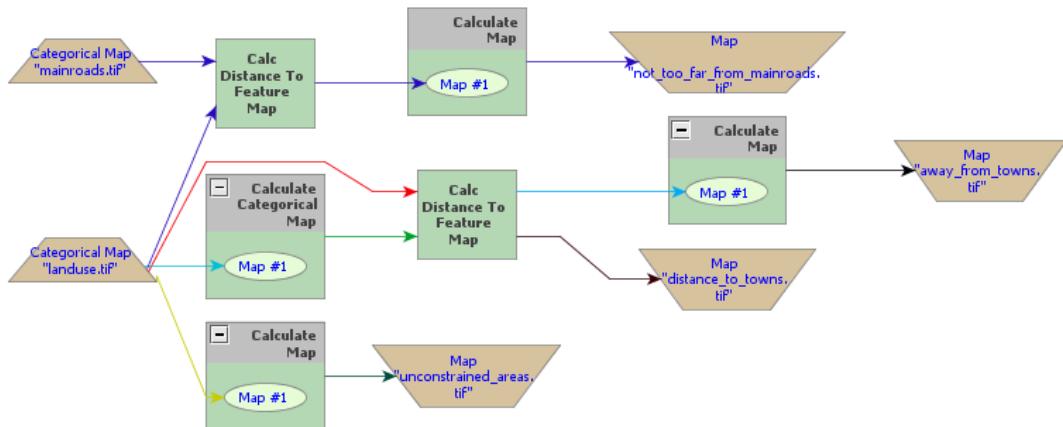
- Distância até estradas principais < 15 quilômetros.
- Distância mínima de 1000 metros até planícies de inundação.

Você precisará carregar outro *Load Categorical Map* e inserir o arquivo “mainroads.tif” da pasta lesson5. Você precisará habilitar a opção **Define Null Value** e mudar **Null Value** para “0”. Você pode conectar este functor diretamente em outro *Calc Distance to Feature Map* que você precisará inserir no plano de sketch. Use o mapa categórico “landuse.tif” novamente como **Mask**.

Use as seguintes equações nos respectivos Calculate Map (você precisará de um para cada critério): **if i1 > 15000 then null else 1**

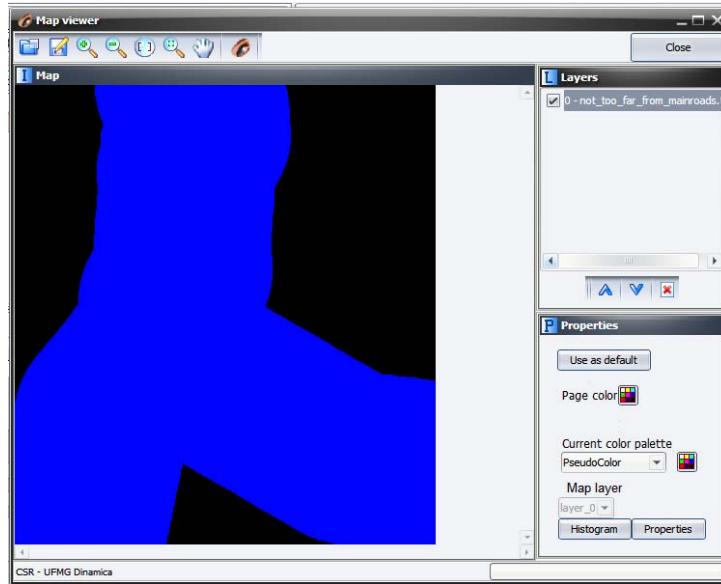
Agora conecte a sua saída no *Save Map* “not_too_far_from_mainroads.tif”

Você obterá o seguinte modelo:



Verifique sua integridade, salve como “my_MCE_part1&2.1.xml” e execute-o.

Você obteve um mapa “not_too_far_from_mainroads.tif” como este?



Para o critério de planícies de inundação, você precisará adicionar outro *Calculate Categorical Map* para apontar as áreas de inundação antes de passar o mapa resultante para o *Calc Distance to Feature Map*.

Conecte o mapa categórico “landuse.tif” a ele e escreva a seguinte equação na caixa de equação do *Calculate Categorical Map*.

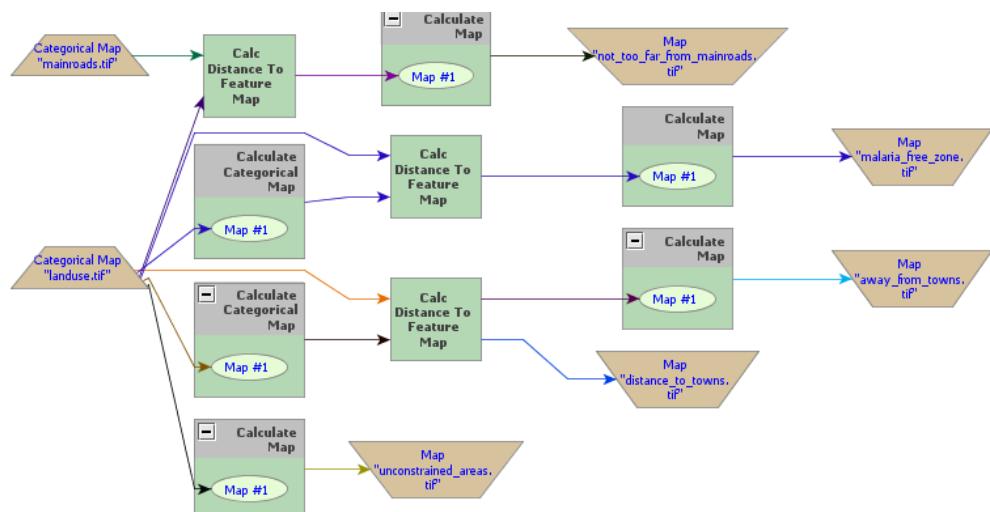
if i1 = 0 then 1 else null

Agora conecte a saída em *Calc Distance to Feature Map* como o porto **Source**, use novamente o mapa categórico “landuse.tif” como o porto **Mask**, em seguida conecte sua saída em outro *Calculate Map* e escreva a seguinte equação:

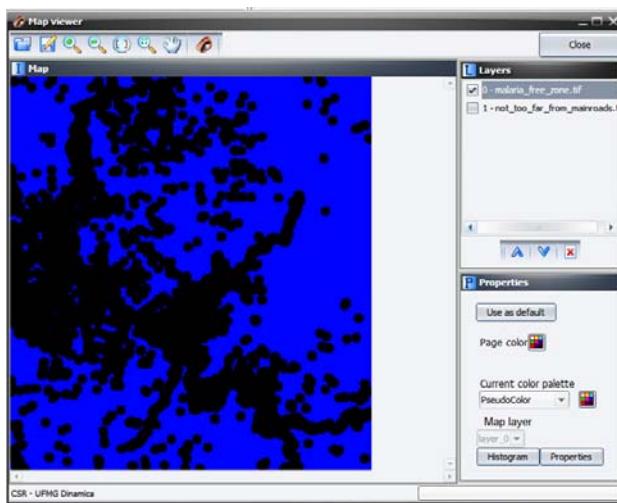
if i1 < 1000 then null else 1

Salve o resultado como “malaria_free_zone.tif”

O modelo para os três critérios combinados deverá se parecer com o modelo abaixo:



Verifique o modelo, salve-o como “my_MCE_part1&2complete.xml”, execute-o e analise o mapa “malaria_free_zone.tif”. Ele se parece com este?



6.3 Terceiro passo: Calculando tempo de deslocamento para cidades vizinhas

O tempo de deslocamento para células vizinhas não deverá exceder a 30 minutos de viagem. Este quarto critério requer o uso do *Calc Cost Map* assim como uma superfície de fricção para o transporte.

Para inserir este critério, estabeleceremos duas velocidades de viagem: uma para a malha rodoviária e outra para todos os tipos de usos da terra. Viajar pelas rodovias permite em média uma velocidade de 60 km/h. Assim, o tempo necessário para atravessar uma célula de 100 metros de

Explicação
$x \text{ (horas)} \rightarrow 0.1 \text{ km}$
$1 \text{ h} \rightarrow 60 \text{ km}$
Isso é igual a $0.1 \text{ km}/60 \text{ km} \times 60 \text{ minutos} = 0.1 \text{ minuto}$

resolução será de 0.1 minuto. Em contrapartida, viajar através de áreas vicinais permite uma velocidade média de 20 km/h, que é equivalente a 0.3 célula/minuto.

Vamos agora reclassificar o mapa “*landuse.tif*” para produzir a superfície de fricção. Para fazer isso você precisará inserir outro *Calculate Map* e escrever em sua caixa de equação a seguinte equação:

if i1 = 6 then 0.1 else 0.3

Agora conecte sua saída a um functor *Calculate Cost Map* (em seu porto **Friction**). Você precisará de outro *Calculate Map* para apontar as cidades vizinhas no mapa de uso da terra usando a equação abaixo:

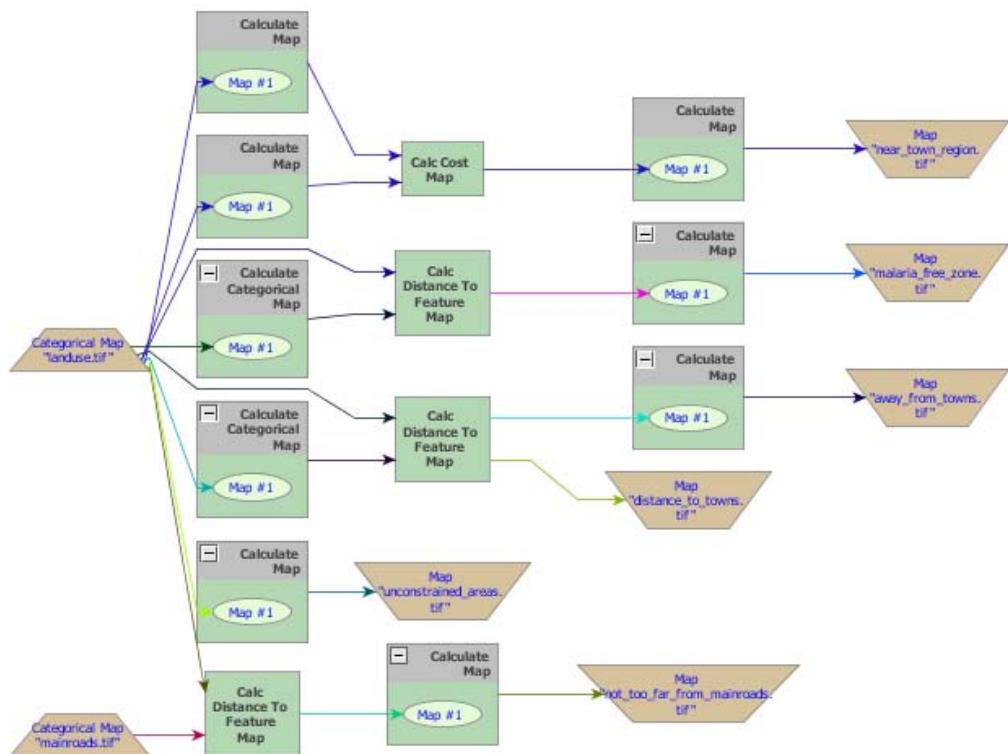
if i1 = 5 then 1 else null

Importante: Não se esqueça de mudar **Data Cell Type** no *Calculate Map* que gera o mapa de fricção e também no *Calc Cost Map* para “IEEE 754 Bit Real”. Lembre-se de que o custo celular é expresso em números fracionários.

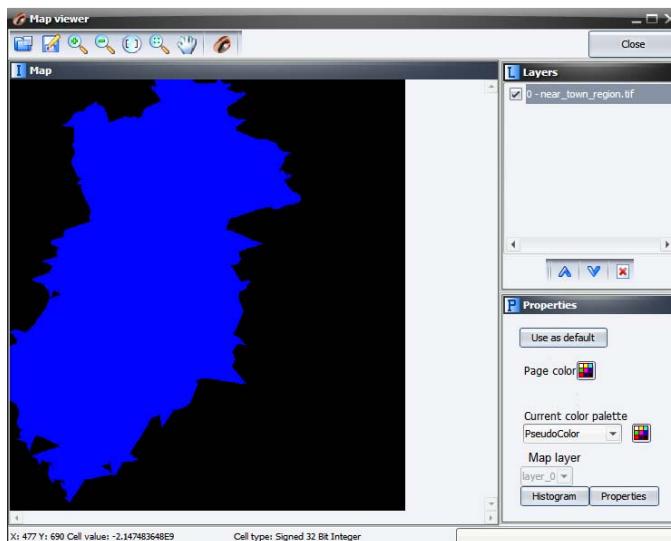
Agora passe a saída do *Calc Cost Map* para outro *Calculate Map* para produzir um mapa booleano de acordo com o tempo máximo de deslocamento, como a seguir:

if i1 < 30 then 1 else null

Por fim, salve o mapa resultante como “near_town_region.tif”. Note que você adicionou ao mapa três functores *Calc Map* adicionais, três functores *Number Map*, um *Calc Cost Map* e outro *Save Map*. O modelo deverá se parecer como a seguinte figura, na qual os functores adicionados nesta etapa aparecem na parte superior.



Verifique a integridade do modelo, salve-o como “my_MCE_part1&2&3.xml”, e execute-o. Examine o mapa “near_town_region.tif”. Foi isto o que você obteve?



6.4 Quarto passo: Combinando os critérios booleanos

Até agora você implementou os seguintes critérios:

- 1) Distância até estradas principais < 15 quilômetros.
- 2) Tempo de deslocamento até cidades vizinhas < 30 minutos.
- 3) Distância mínima de 10 km até cidades existentes.

- 4) Distância mínima de 1000 metros até planícies de inundaçāo.
- 5) Somado a isso, temos que considerar que nem todos os terrenos estão disponíveis. Nós não queremos encorajar mais desmatamento na região, portanto, precisamos usar apenas áreas já desmatadas ou abandonadas. Também devemos excluir áreas urbanas já existentes, rios e planícies de inundaçāo.

Os dois critérios restantes (áreas iguais ou maiores que 1000 hectares, e declividade média de < 0.5 graus) devem ser implementados usando uma abordagem de zonas como aquela fornecida pelos functores do grupo Region. Antes de iniciar esta etapa, você precisa combinar todos os critérios já implementados anteriormente em um único mapa.

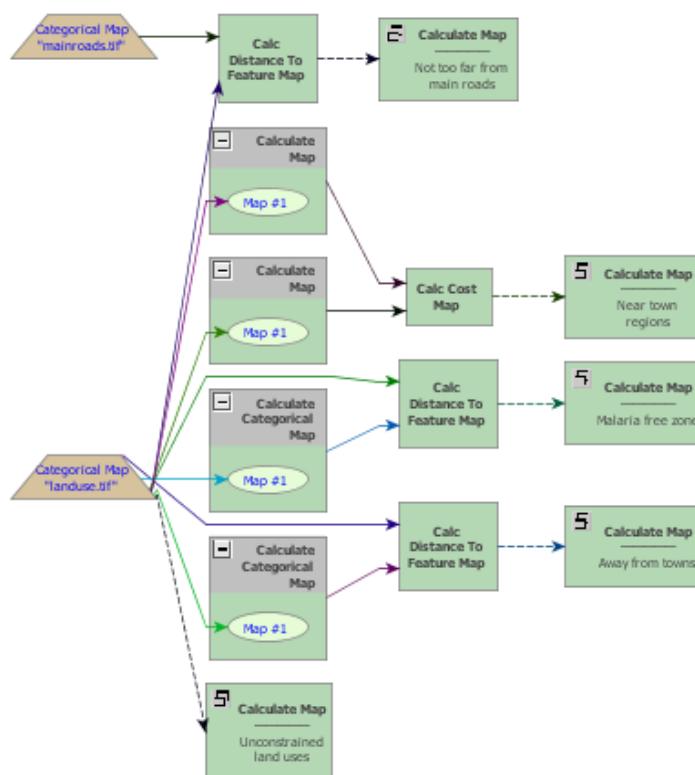
Como você já percebeu este modelo atingiu certo nível de complexidade, e está começando a ficar difícil de compreender sua estrutura através da interface gráfica. Dinamica EGO oferece alguns recursos para reduzir a complexidade visual de modelos.

Primeiramente, você pode apagar os functores *Save Map*, uma vez que você já verificou seus resultados. Em segundo lugar, você pode adicionar comentários a alguns functores destacando seus produtos. Façamos isso.

Arraste a ferramenta *Add Comment to Functor* da barra de ferramentas do sketch. Descreva o produto de cada functor que precede os functores *Save Map*, assim você pode traçar cada fluxo de dados até seu ponto inicial (que é o mapa “landuse.tif”). Para conseguir ver o comentário que você adicionou é necessário fechar o container *Calculate Map* primeiro. Faça isso clicando em seu ícone superior esquerdo.

Escreva o seguinte nos respectivos functores: “Near town regions”, “Malaria free zone”, “Away from towns”, “Unconstrained land uses”, “Not to far from main roads”.

Neste ponto, você pode apagar os functores *Save Map*, uma vez que você já verificou seus resultados. Depois disso, organize o layout do modelo da esquerda para a direita. **DICA:** ajuste o tamanho da janela sketch para obter uma visão mais detalhada do modelo. Isto é o que você obteve:



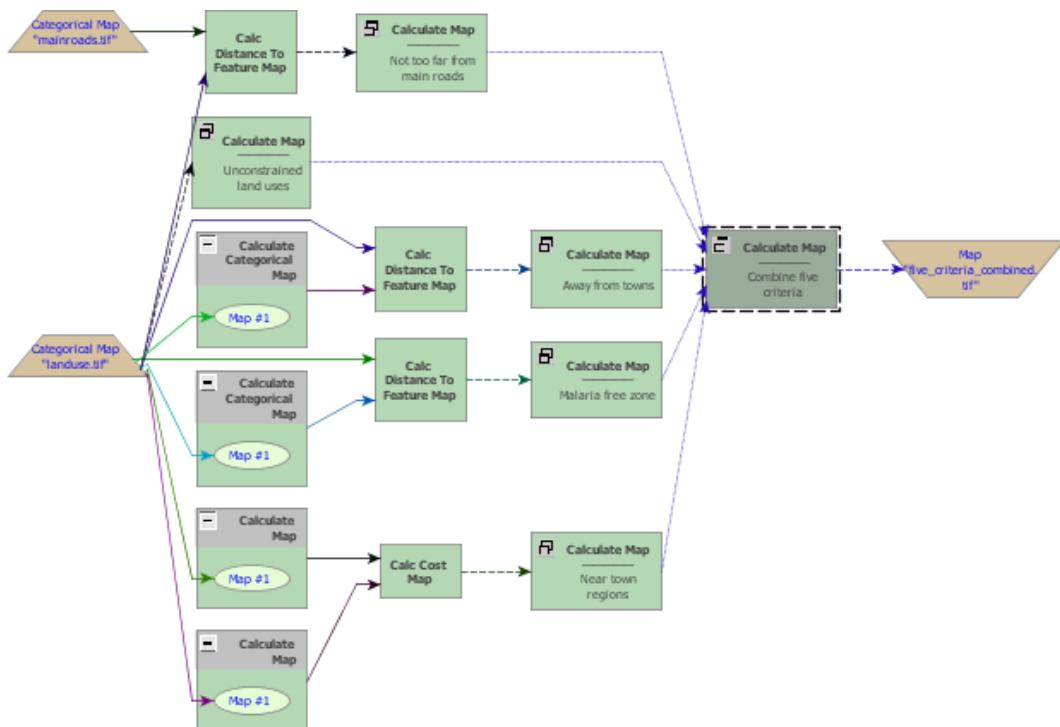
Os cinco functores *Calculate Map* produzem os cinco critérios resolvidos anteriormente. Agora você precisa combiná-los com um *Calculate Map*.

Agora adicione cinco functores *Number Map* dentro dele. Abra-o e escreva a seguinte equação:

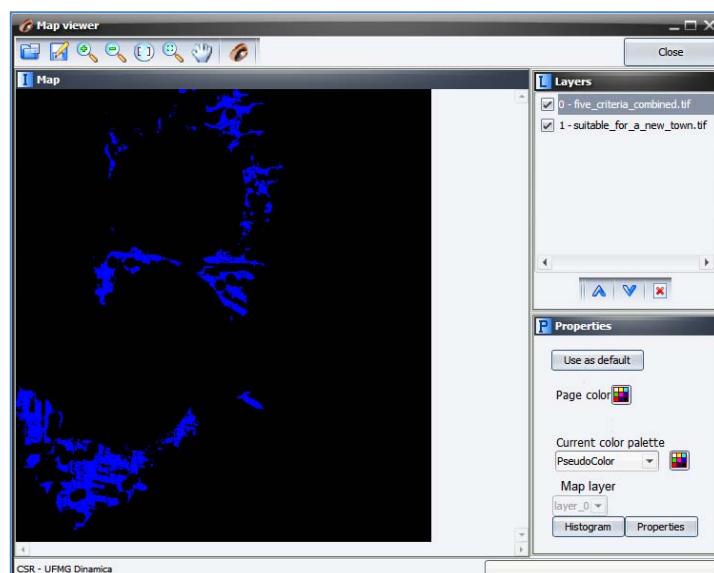
$$i1*i2*i3*i4*i5$$

DICA: *Calculate Map* não processa células nulas.

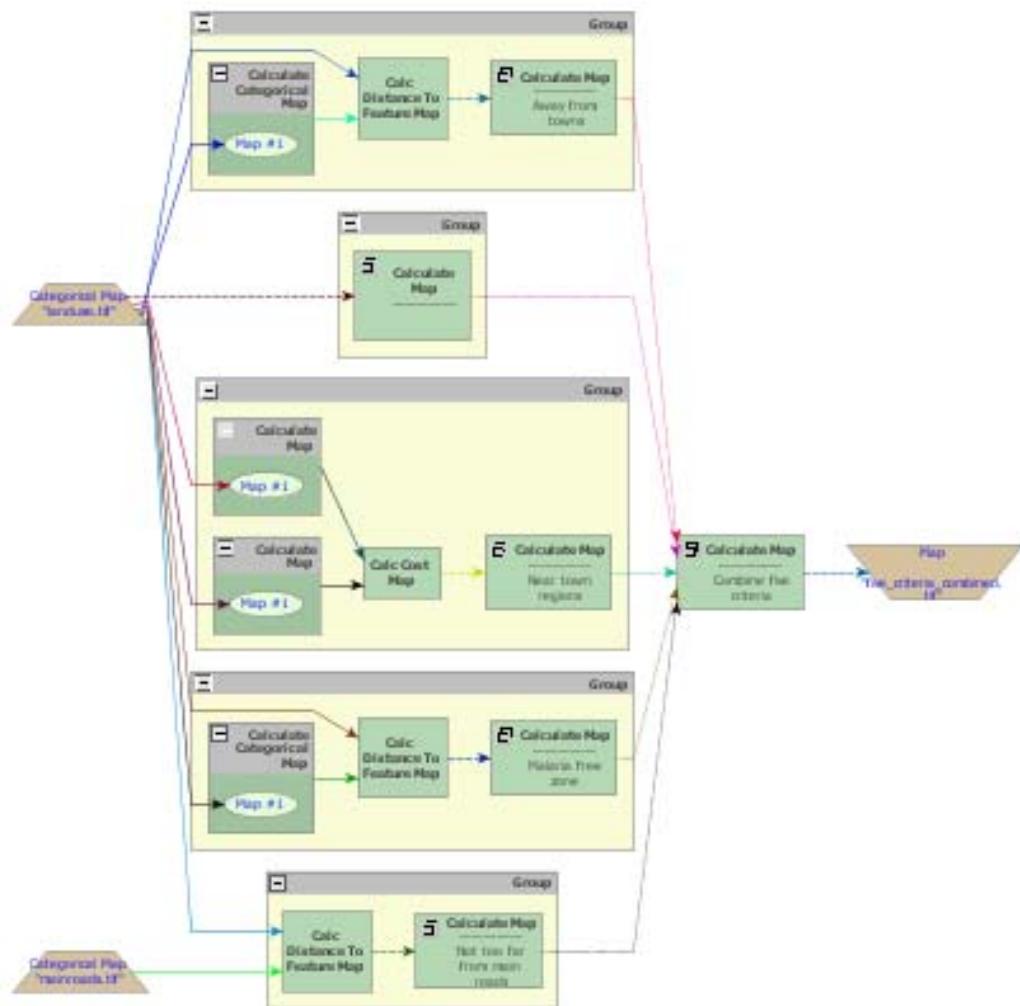
Mude o **Data Cell Type** para “Unsigned 8 Bit Integer” e **Null Value** para “0”; escreva o comentário “Combine five criteria” e adicione um *Save Map* para ver o resultado, nomeando o arquivo como “five_criteria_combined.tif”. Por fim, organize o layout da esquerda para a direita.



Feche o primeiro container à esquerda para mostrar os comentários, salve o modelo como “my_MCE_part1&2&3&4.xml”, verifique o modelo e execute-o. É isto o que você obteve?



Para simplificar mais ainda o visual do modelo, outro recurso disponível no Dinamica EGO é o functor *Group*. Arraste cinco functores *Group* da aba *Control* e coloque-os no plano de sketch. Agora selecione cada functor do fluxo de dados que termina no *Calculate Map* “Combine five criteria” (Pressione **Crt + botão direito** para selecionar mais de um functor), deixe fora os functores *Categorical Map*, e então arraste os functores selecionados para um *Group*. Repita este procedimento para os cinco critérios.



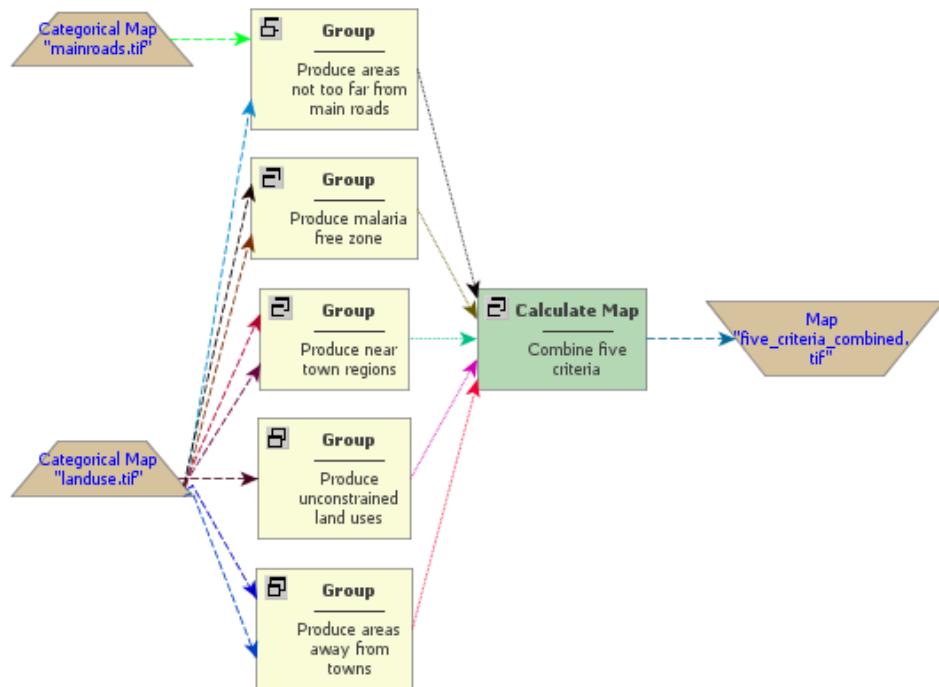
Feche cada *Group* e escreva os seguintes comentários em seus respectivos functores *Group*. Talvez você precise ampliar a visualização do modelo.

- Produce near town regions
- Produce malaria free zone
- Produce areas away from town
- Produce unconstrained land uses
- Produce areas not too far from main roads

Veja o modelo na próxima página. O modelo não se tornou mais fácil de visualizar? É claro, você precisará abrir o functor *Group* para entender como os dados são produzidos. Você pode fazer isso um por um. Veja que a função do functor *Group* é ajudar a organizar o diagrama do modelo.

DICA: *Group* é também usado para assegurar a ordem certa de execução, dado que ele sempre executa os functores agrupados antes dos functores subsequentes.

Salve este modelo como “my_MCE_part1&2&3&4group.xml” e vamos agora para o próximo passo.

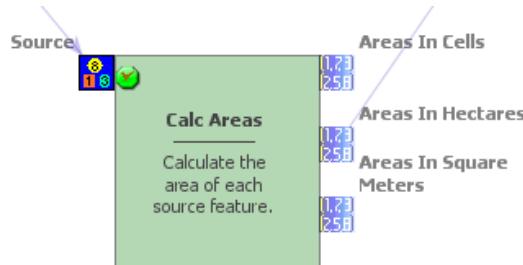


6.5 Quinto passo:

Nesta etapa, você irá identificar as áreas adequadas iguais ou maiores que 1000 hectares. Primeiro, apague o mapa “five_criteria_combined.tif”. Lembre-se que quanto menos o programa acessa o disco, mais rápido o modelo roda. Agora, vá até a aba Map Algebra e pegue o functor *Calc Patch Label Map*. Este functor rotula manchas de células usando números sequenciais. Uma mancha consiste em um grupo de células vizinhas de mesma categoria (veja também lição 10: Métricas de paisagem no Dinamica EGO). Você precisará disso para implementar o critério relativo à área. Conecte a saída de “Combine Five Criteria” a esse functor. Alguns desses parâmetros são:

Initial Patch Label: Insira “1”. **Importante:** Ajuste **Null Value** para “0” e **Data Cell Type** para **Signed 32 Bit Integer**.

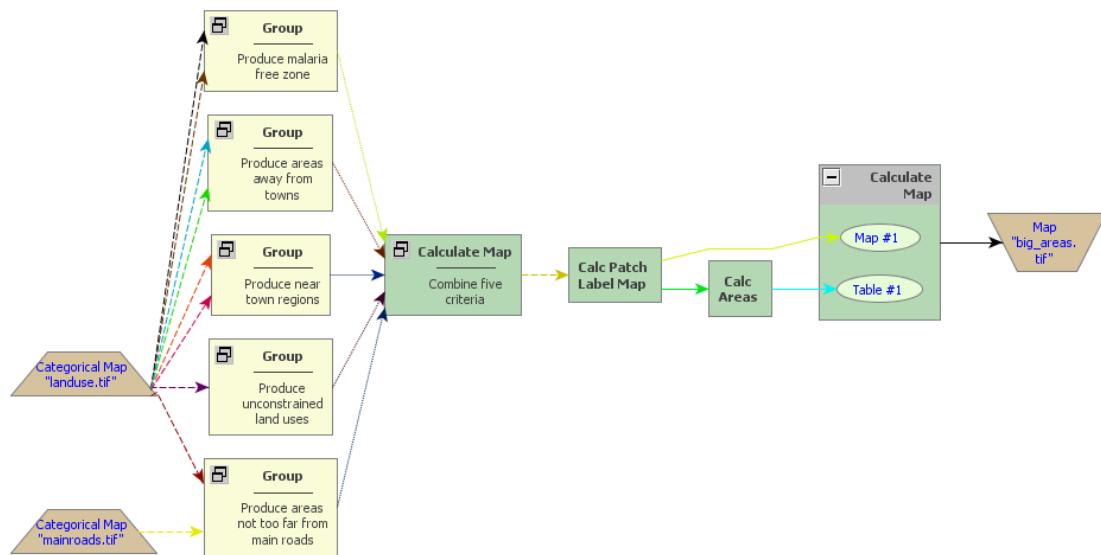
Isso será o suficiente para armazenar todos os identificadores de manchas. Deixe os demais parâmetros sem mexer. Para maiores detalhes acesse o Help do Dinamica EGO. Agora conecte a saída de *Calc Patch Label Map* a *Calc Areas*, disponível na aba Map Algebra. Como saída este functor dispõe de 3 opções: Nós precisamos da área em hectares, então usaremos o porto **Areas in Hectares** para a próxima conexão.



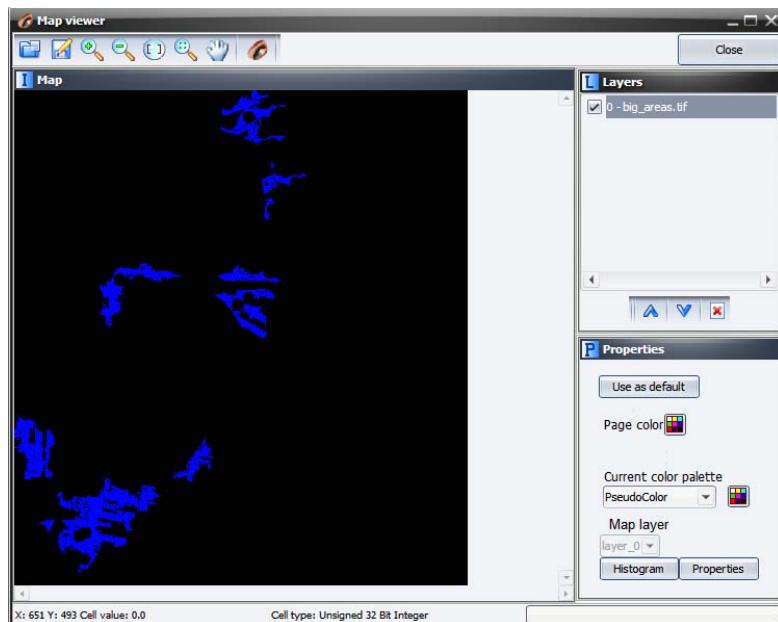
Agora, coloque no plano de sketch um *Calculate Map*, um *Number Map* e um *Number Table* dentro dele. Conecte *Calc Areas* a *Table 1* (isto implica que a saída de *Calc Areas* é uma tabela), e a saída *Map* de *Calc Patch Label Map* a *Map #1*. Agora abra *Calculate Map* e escreva:

if (t1[i1] < 1000) then null else 1

Usando essa fórmula, o functor irá analisar o tamanho de cada mancha e depois eliminar as manchas menores que 1000 hectares. Agora insira um *Save Map* e nomeie o arquivo como "big_areas.tif".

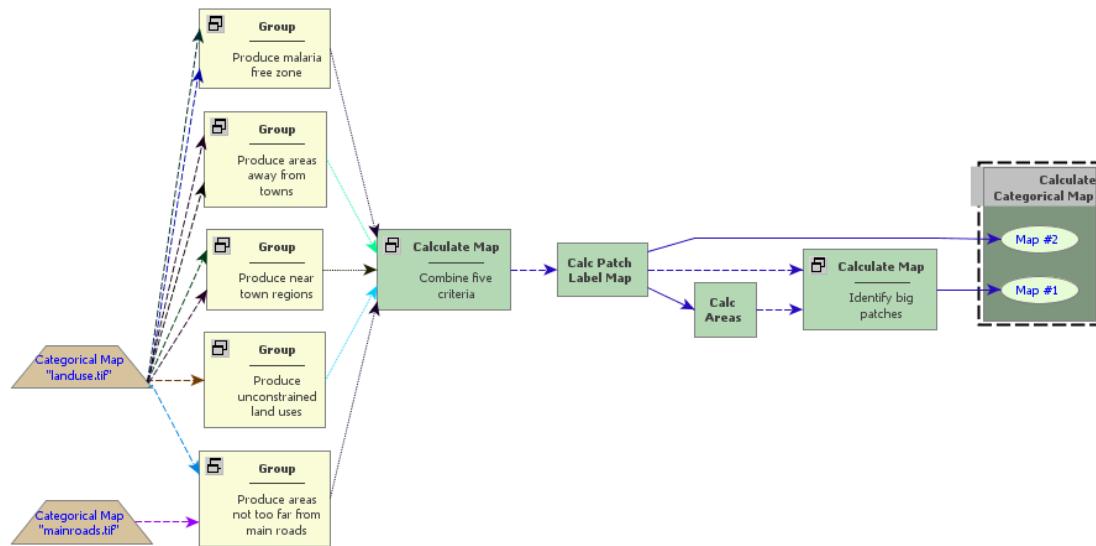


Verifique o modelo, salve-o como "my_MCE_part1&2&3&4&5.xml", execute-o e examine seu resultado. É isto o que você obteve?



Ótimo! Isso foi custoso! Mas antes de nos movermos para o próximo passo, ainda há uma última coisa a fazer. Para manter os identificadores das manchas, você precisa multiplicar o resultado do último *Calculate Map*, no qual você adicionou o comentário "Identify big patches", pela a saída do *Calc Patch Label*. Assim, você manterá os identificadores apenas para as manchas grandes.

Insira um *Calculate Categorical Map* e dois functores *Number Map* dentro dele. Ajuste seus *Number Map* para “1” e “2” respectivamente, e remova o functor *Save Map*. Agora conecte os functores da seguinte forma:



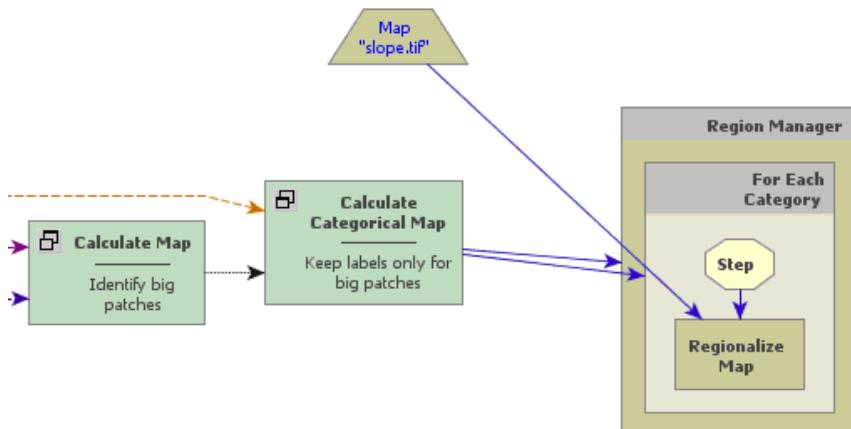
DICA: Você precisará de um *Calculate Categorical Map* ao invés do *Calculate Map* porque os rótulos das manchas serão usados em seguida para regionalizar o mapa de declividade. Ajuste **Null Value** para “0” e **Cell Type** para **Signed 32 Bit Integer**.

Escreva: **i1*i2**, feche o functor e insira o comentário “keep labels only for big patches”. Agora você está pronto para ir para o último passo.

6.6 Sexto passo:

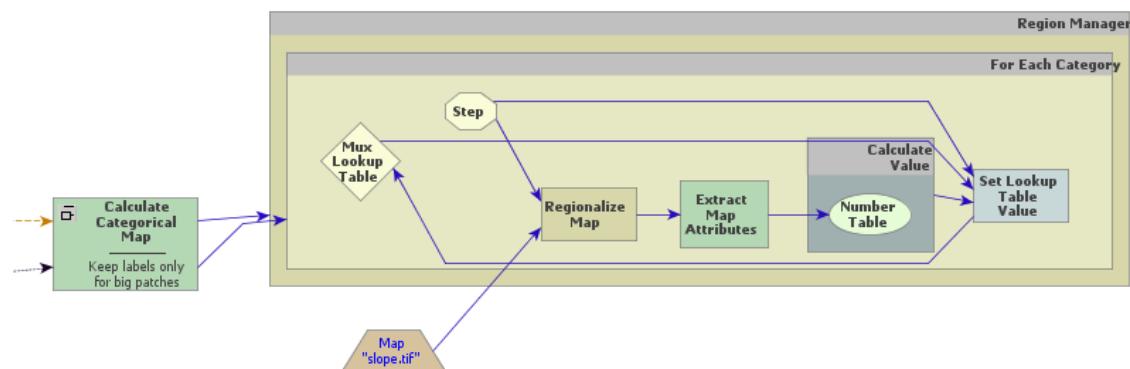
O último critério estabelece que a declividade média nas manchas selecionadas deve ser menor que 0.5 grau. Insira um *Load Map* e carregue o arquivo “\lesson4\originals\slope.tif”. Agora revise a Lição 3, pois você precisará aplicar novamente o conceito de região.

Arraste para o plano de sketch um *Region Manager*, *For Each Category*, *Regionalize Map*, e um *Step* e insira os últimos dois functores dentro do primeiro e *For Each Category* dentro do *Region Manager* e conecte-os da seguinte maneira:

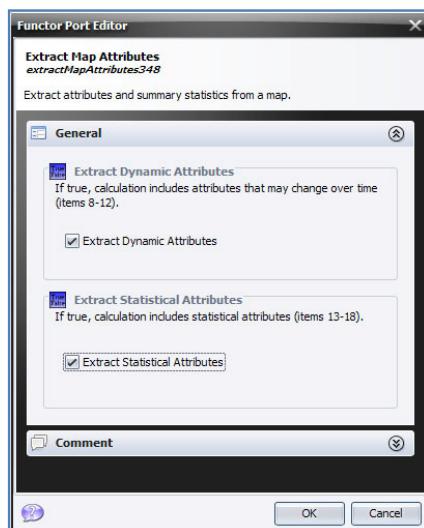


O mapa a ser regionalizado é “slope.tif”. O mapa de saída do último *Calculate Categorical Map* irá controlar ambos os *Region Manager* e *For Each Category*. Lembre-se que este functor permite que o modelo itere de acordo com as categorias de mapas passados a ele. Agora insira

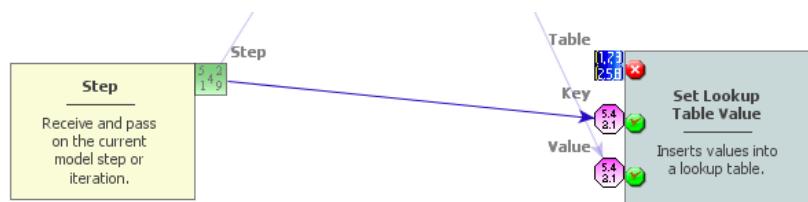
um *Mux Lookup Table*, um *Extract Map Attributes*, um *Calculate Value*, um *Number Table* e um *Set Lookup Table*, todos dentro de *For Each Category* e conecte-os como a seguir:



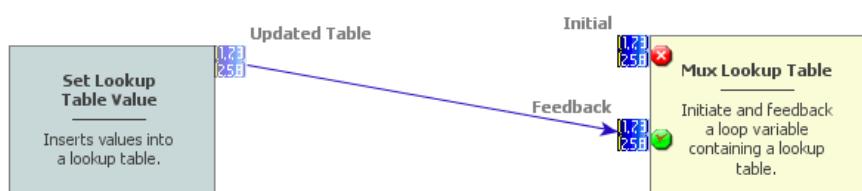
Escreva a equação: **t1[13]** no *Calculate Value*. Lembre-se que 13 é a chave para a média dos valores de todas as células não nulas na tabela de atributos. Este submodelo é muito parecido com aquele introduzido na Lição 3. Abra *Extract Map Attributes* e habilite a opção:



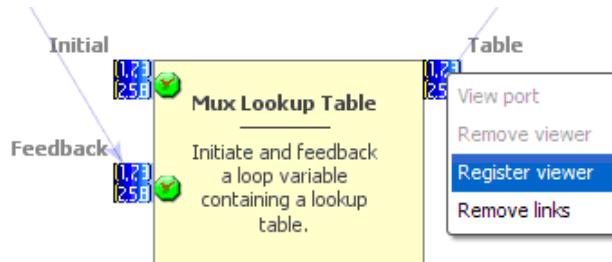
Lembre-se de ligar *Step* a **Key** no *Set Lookup Table*,



assim como **Updated Table** ao porto **Feedback**.



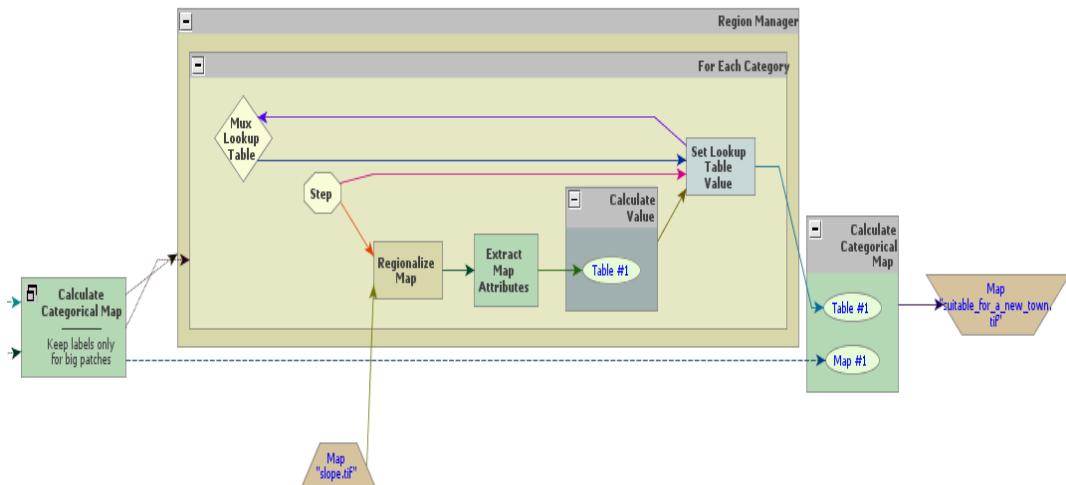
Abra um *Mux Lookup Table* com o Edit Functor Ports e clique no **Initial**, insira uma tabela em branco. Para isso, apenas insira um “0” e “0”, como **Key** e **Value**. Selecione a opção Register viewer no porto de saída **Table**. Assim, você poderá ver os resultados.



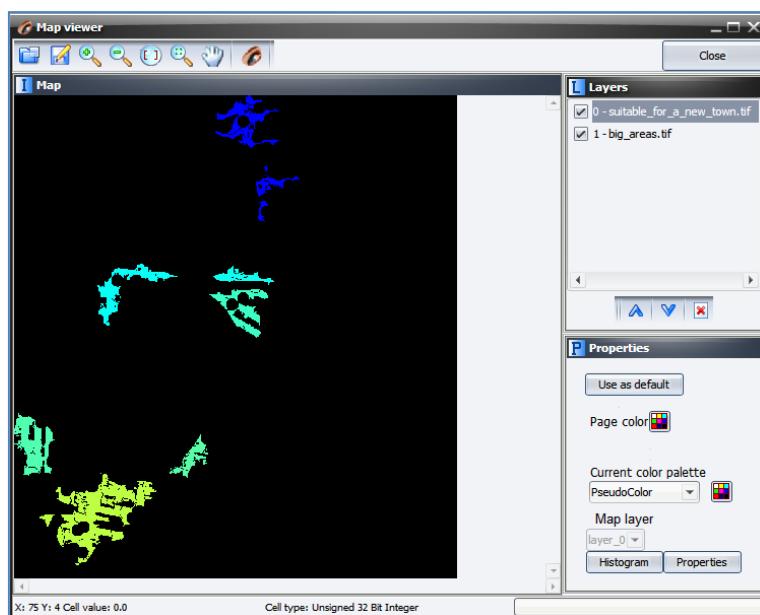
Por fim, insira um *Calculate Categorical Map* depois do *Region Manager* e coloque um *Number Map* e um *Number Table* dentro dele. Numere-os e escreva a seguinte equação:

if t1[i1] > 0.5 then null else i1

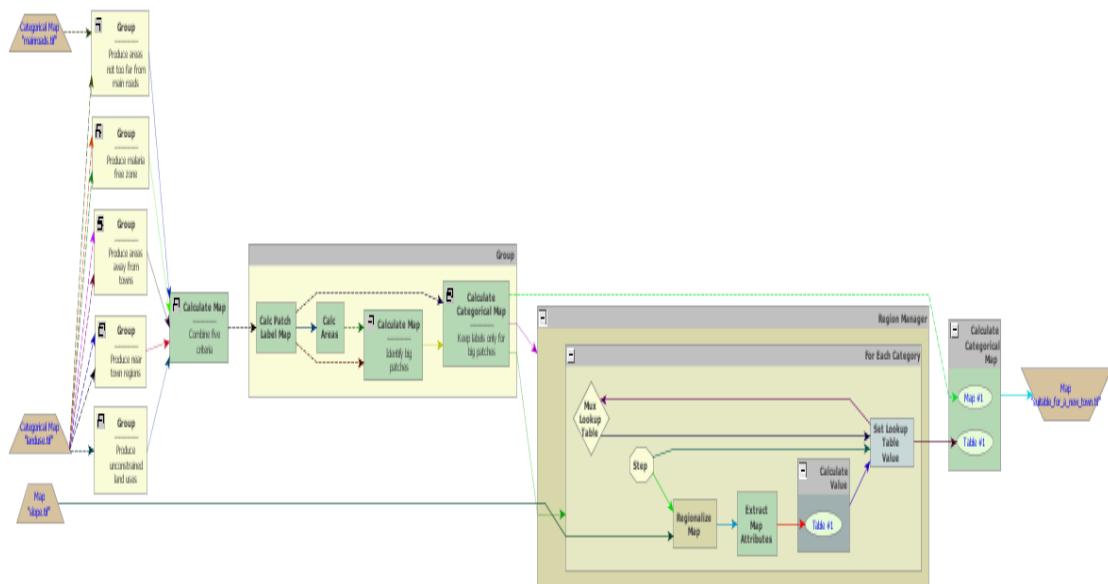
Salve o resultado como “suitable_for_a_new_town.tif”.



Verifique o modelo, salve-o como “my_MCE_part1&2&3&4&5&6.xml” e execute-o. É isto o que você obteve?

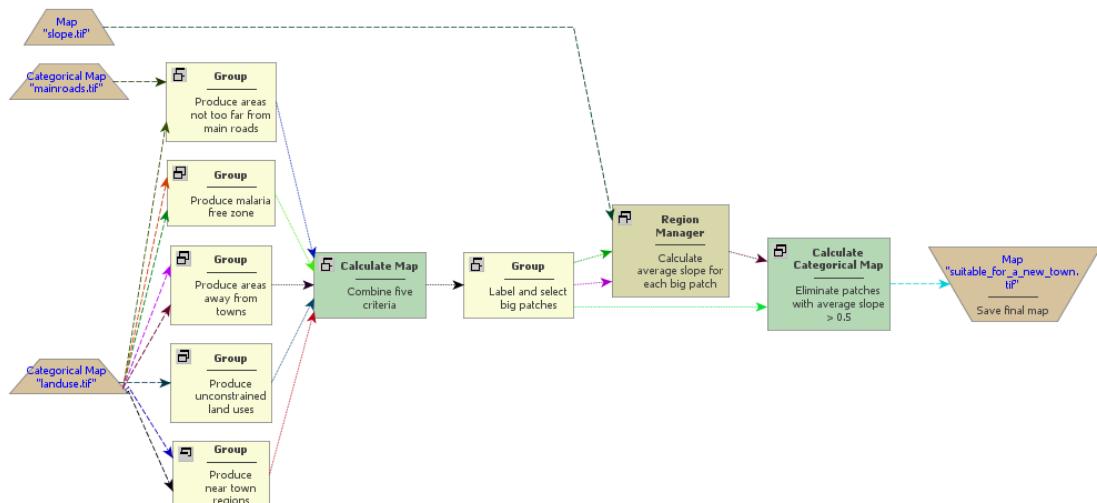


Por fim, você só precisa organizar o modelo em um layout mais legível. Insira um *Group* e arraste os seguintes funtores para dentro dele: *Calc Patch Label Map*, *Calc Areas*, *Calculate Map* “Identify big areas” e *Calculate Map* “Keep labels only for big patches”.



Feche o functor *Group* e adicione o comentário: “Label and select big patches”, feche o *Region Manager* e escreva o comentário: “Calculate average slope for each big patch”, feche o último functor *Calculate Map* e escreva “Eliminate patches with average slope > 0.5” como comentário e por último, escreva “Save final map” ao *Save Map* “suitable_for_a_new_town.tif”.

Salve o modelo como “my_MCE_part1&2&3&4&5&6_complete.xml”. Este é o modelo final:



DEVER DE CASA

Ao examinar o mapa “suitable_for_a_new_town.tif”, você irá notar uma grande mancha na sua porção sudoeste, a qual será o melhor alvo para uma nova cidade, considerando a área de superfície como um critério adicional. Você seria capaz de adicionar um submodelo ao modelo atual para identificar esta mancha?

DICA: Você precisará recalcular as áreas para as manchas restantes e passar essa tabela ao *For Each*, que permite uma varredura pela tabela. Adicione dois *Mux Value* dentro de *For Each*, um que irá armazenar o valor máximo de área e outro que irá armazenar sua chave.

7. Lição 6: Construindo um modelo de simulação de mudanças de uso e cobertura da terra

O que você irá aprender?

- Como simular mudanças no uso da terra
- Como calcular matrizes de transição
- Como calibrar um modelo de mudanças no uso da terra
- Uso dos Pesos de Evidências
- Correlação de mapas
- Validação de modelos
- Functores:
 - *Determine Weights Of Evidence Ranges*
 - *Determine Weights Of Evidence Coefficients*
 - *Calc WOfE Probability Map*
 - *Calc Change matrix*
 - *Patcher*
 - *Expander*

O desenvolvimento de modelos espaço-temporais, nos quais o estado ou o atributo de certa localização geográfica muda com o passar do tempo em resposta a um conjunto de forças motoras, é um pré-requisito crucial para a modelagem ambiental e, portanto, abre uma vasta gama de possibilidades para a representação dos fenômenos dinâmicos.

Nesse contexto, este exercício explora o uso do Dinamica EGO como uma plataforma de simulação para modelos de mudanças no uso e cobertura da terra (LUCC). O objetivo é calibrar, executar e validar um modelo LUCC e, neste caso, trataremos de um modelo de simulação do desmatamento. Você precisará passar por 10 passos para completar o modelo, como representado na fig. 8. Para facilitar este processo, cada um desses passos será representado como um modelo separado. Embora todos os passos possam ser unidos em um único modelo, por razões de simplicidade nós os manteremos como modelos separados.

O conjunto de dados de entrada representa uma região do estado de Rondônia, nos arredores da cidade de Ariquemes, na Amazônia Brasileira (fig. 9). Abra os mapas “23267_1997.ers” e “23267_2000.ers”, da pasta “\Examples\setup_run_and_validade_a_lucc_model\originals”, usando **Color Palette** “Amazon”. Estes mapas correspondem a uma imagem Landsat (232/67) classificada pelo PRODES (INPE, 2008) – Monitoramento da Floresta Amazônica brasileira por satélite – para os anos 1997 e 2000.

Neste modelo LUCC, Dinamica EGO irá usar o mapa de 1997 como paisagem inicial e o mapa de 2000 como paisagem final, considerando a paisagem como uma matriz bi-dimensional de tipos de usos da terra.

Os mapas de paisagem possuem as seguintes classes; o valor nulo é representado pelo 0:

Chave	Classes de Uso da Terra
1	Desmatado
2	Floresta
3	Cerrado

DICA: Tenha em mente que os números é que identificam as classes dos mapas, uma vez que Dinamica EGO não armazena nomes de classes explicitamente.

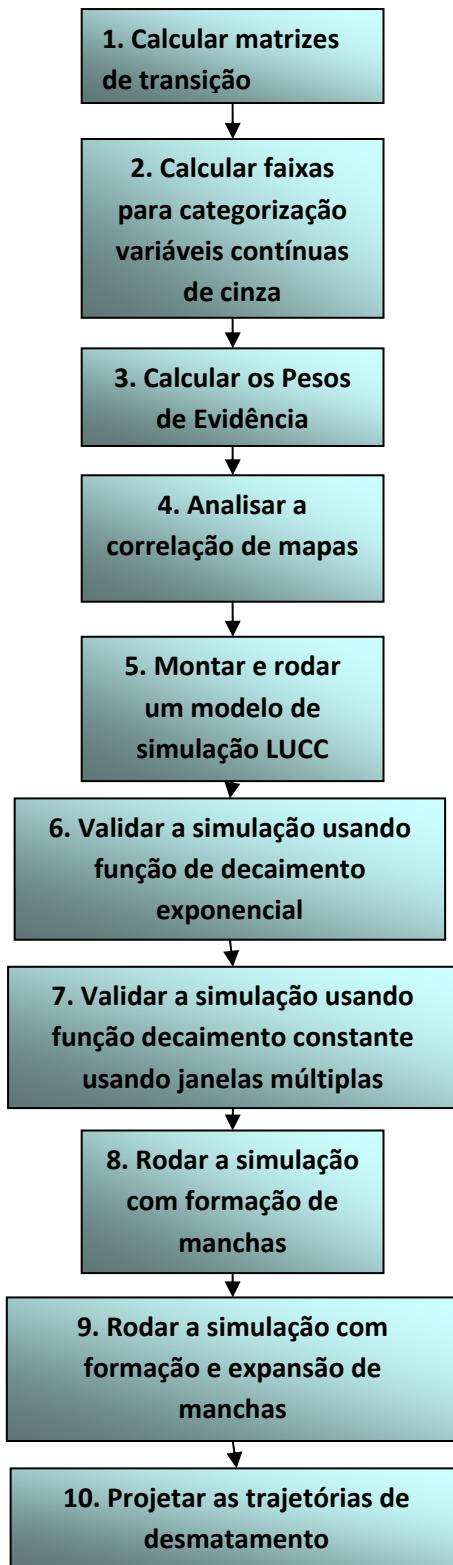


Fig. 8 – Dez passos do modelo de simulação de mudanças no uso e cobertura da terra desenvolvido na lição 6.

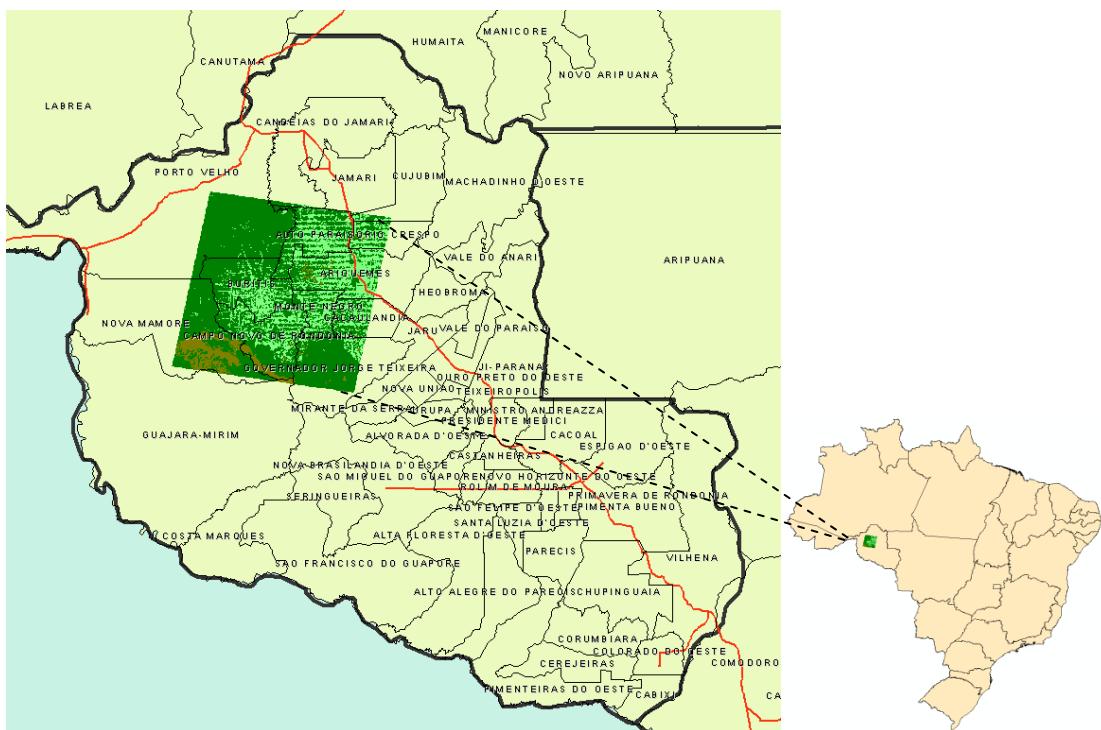


Fig. 9 – Área de estudo em relação ao estado de Rondônia e ao Brasil (verde escuro é floresta, verde claro é área desmatada, e marrom é cerrado).

7.1 Primeiro passo: Calculando matrizes de transição

Primeiro, você precisará calcular as matrizes de transições históricas . A matriz de transição descreve um sistema que muda em intervalos discretos de tempo, nos quais o valor de qualquer variável em um dado período de tempo é a soma das porcentagens fixas dos valores de todas as variáveis do passo de tempo precedente. A soma das frações ao longo da coluna da matriz de transição é igual a 1 (eq. 1). A linha diagonal da matriz de transição não precisa ser especificada uma vez que Dinamica EGO não modela a porcentagem de células inalteráveis, nem as transições iguais a zero. A taxa de transição pode ser passada ao modelo LUCC como um parâmetro fixo ou ser atualizada pela retroalimentação do modelo.

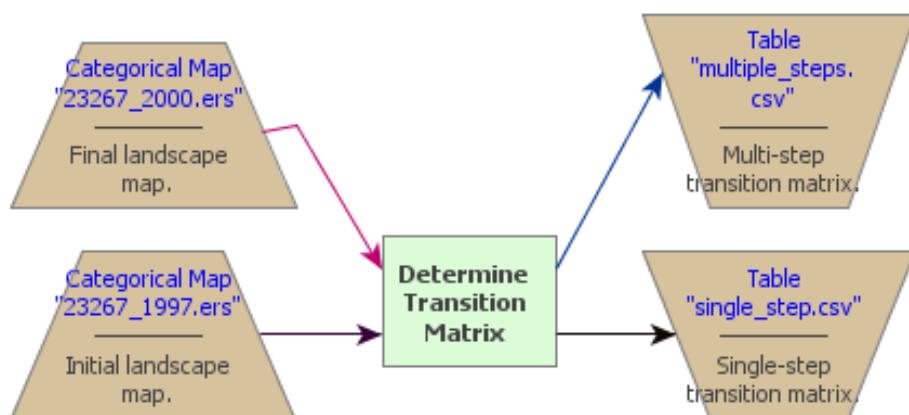
$$\begin{bmatrix} 1 \\ 2 \\ \cdot \\ j \end{bmatrix}_{t=v} = \begin{bmatrix} P_{11} & P_{12} & P_{1\cdot} & P_{1j} \\ P_{21} & P_{22} & P_{2\cdot} & P_{2j} \\ P_{31} & P_{32} & P_{33} & P_{3j} \\ P_{j1} & P_{j2} & P_{j\cdot} & P_{jj} \end{bmatrix}^v * \begin{bmatrix} 1 \\ 2 \\ \cdot \\ j \end{bmatrix}_{t=0} \quad (1)$$

A matriz de passo único corresponde a um período de tempo representado como um único passo de tempo, por sua vez, a matriz de passos múltiplos corresponde a uma unidade de passo de tempo (ano, mês, dia, etc.) especificado pela divisão do período de tempo por um número de passos de tempo. Para Dinamica EGO, o passo de tempo pode compreender qualquer extensão de tempo, uma vez que as unidades de tempo são apenas uma referência externa. Uma matriz de transição de múltiplos-passos pode apenas ser derivada de uma matriz ergódica, isto é, uma matriz que possui auto-valores e auto-vetores reais.

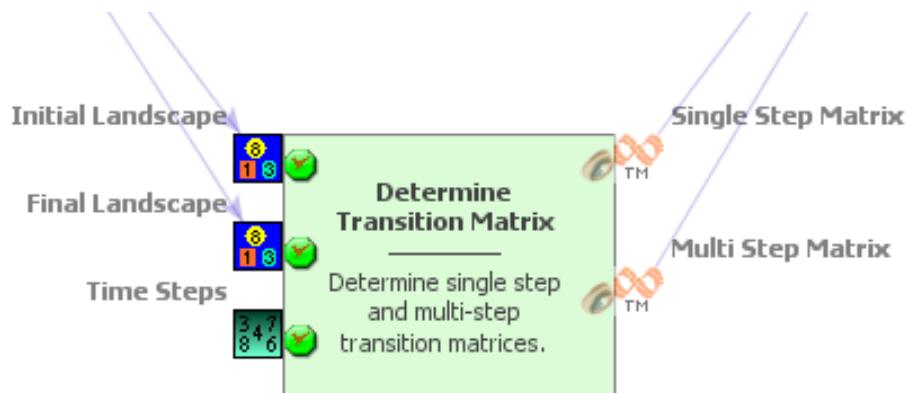
As taxas de transição determinam a quantidade líquida de mudanças, isto é, a porcentagem de terra que irá alterar para outro estado (atributo dos usos e cobertura da terra), e são, portanto, conhecidas como taxas líquidas, sendo adimensionais. Em contrapartida, as taxas brutas são especificadas como uma unidade de área, tais como hectares ou km^2 por unidade de tempo. No caso de não haver solução para a matriz de transição de múltiplos passos, você pode rodar o modelo em vários passos de tempo, como definido acima, calculando uma taxa bruta fixa para cada passo de tempo (e.g. ano) dividindo a mudança acumulada no período pelo número de passos que compõem o período (isto não deve ser usado em modelos de transição complexos). Dinamica EGO converte taxas brutas em taxas líquidas, dividindo a extensão da mudança pela fração de cada classe de uso e cobertura anterior à mudança, antes de passá-las para os funtores de transição: *Patcher* e *Expander*.

Abra o modelo “determine_transition_matrix.xml” localizado em
“\setup_run_and_validade_a_lucc_model\1_transition_matrix_calculation”

Este modelo calcula matrizes de passo único e de múltiplos passos.

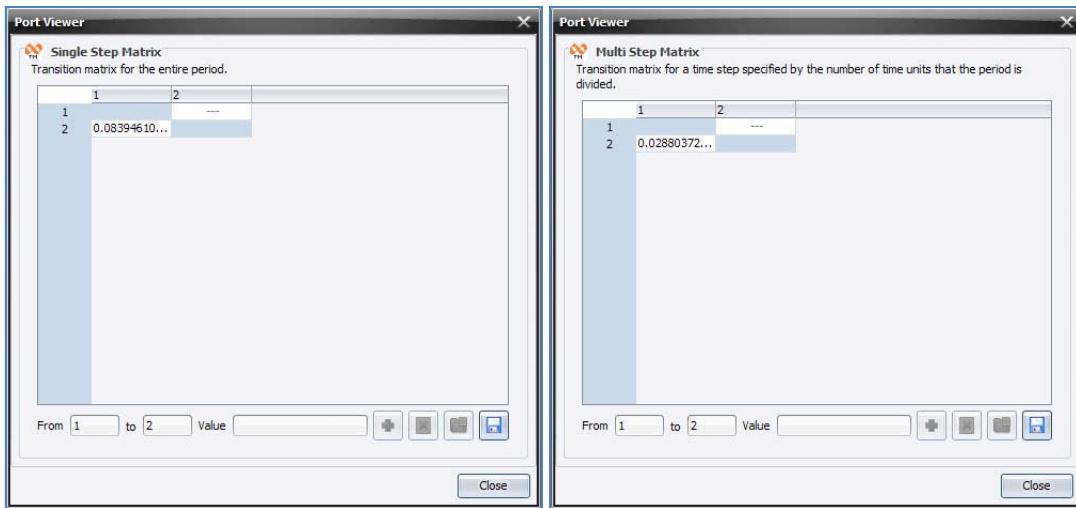


Abra o functor *Determine Transition Matrix* com Edit Functor Ports. **DICA:** Este functor, junto com outros aplicados para calibrar o modelo, é encontrado na aba Calibration.



Observe que o *Categorical Map* “2367_1997.ers” se conecta ao porto **Initial Landscape** e o “2367_2000.ers” ao porto **Final Landscape**. Verifique o número de passos de tempo. Neste caso, insira “3”, uma vez que você quer determinar a matriz de múltiplos passos por passos anuais (2000 – 1997 = 3 anos).

Habilite a opção **Register viewer** nos portos de saída e execute o modelo. Abra as matrizes resultantes clicando com o botão direito nos portos de saída. É isto o que você obteve?



Você pode também ver os resultados do modelo na janela de log. Você precisa apenas rolar a barra de volta. **DICA:** você pode copiar os resultados da janela de Log e colá-los como a seguir:

Single Step Transition Matrix:

From \ To	1	2	3
1	XXXX	--	--
2	0.0839461	XXXX	--
3	--	--	XXXX

Multi Step Transition Matrix:

From \ To	1	2	3
1	XXXX	--	--
2	0.0288037	XXXX	--
3	--	--	XXXX

A única transição que está acontecendo é de floresta (2) para área desmatada (1). As taxas indicam que uma porcentagem da floresta está se alterando para área desmatada por unidade de passo de tempo, que é 3 anos para a primeira matriz e 1 ano para a última. Portanto, neste período de tempo, o desmatamento está acontecendo a uma taxa líquida de 2.8 % por ano, o que significa que a floresta remanescente está diminuindo em 2.8% por ano. Tenha em mente que assim como taxas de juros, as taxas de transição são reaplicadas novamente a cada passo de tempo sobre a variável estoque, que neste caso é representada pela extensão de floresta remanescente.

A matriz de transição líquida é passada ao modelo de simulação e Dinamica EGO percorre o mapa de paisagem (uso e cobertura) para contar o número de células para calcular a taxa bruta em termos de quantidade de células a serem alteradas. Para atribuir uma taxa bruta constante você precisará passar uma taxa líquida variável ao modelo, o que também é possível devido à habilidade do Dinamica EGO de incorporar retroalimentação à simulação. Vamos para o próximo passo.

7.2 Segundo passo: Calculando intervalos para categorização variáveis contínuas

O método dos Pesos de Evidência (Goodacre *et al.* 1993; Bonham-Carter, 1994) é aplicado no Dinamica EGO para produzir um mapa de probabilidades de transição (fig. 10), o qual representa as áreas mais favoráveis para uma mudança (Soares-Filho *et al.* 2002, 2005).

Pesos de Evidência consistem em um método Bayesiano, no qual o efeito de uma variável espacial em uma transição é calculado independentemente de uma solução combinada. Os Pesos de Evidência representam cada influência sobre uma variável na probabilidade espacial de uma transição $i \Rightarrow j$ e são calculados da seguinte forma:

$$O\{D|B\} = \frac{P\{D|B\}}{P\{\overline{D}|B\}} \quad (2)$$

$$\log\{D|B\} = \log\{D\} + W^+ \quad (3)$$

Onde W^+ é o Peso de Evidência da ocorrência do evento D , dado um padrão espacial B . A probabilidade a posteriori de uma transição $i \Rightarrow j$, dado um conjunto de dados espaciais (B, C, \dots, N), é expressa da maneira a seguir:

$$P\{i \Rightarrow j | B \cap C \cap D \dots \cap N\} = \frac{e^{\sum W_N^+}}{1 + e^{\sum W_N^+}} \quad (4)$$

Onde B, C, D , e N são os valores das variáveis espaciais k que são medidas na localização x, y e representadas por seus pesos W_N^+

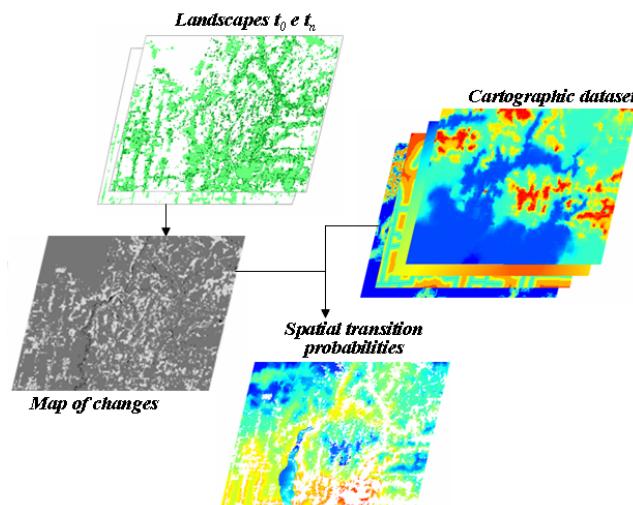


Fig. 10. Cálculo e aplicação dos Pesos de Evidência para produzir um mapa de probabilidades de transição.

Uma vez que os Pesos de Evidência são aplicados somente a dados categóricos, é necessário categorizar os mapas de variáveis contínuas (dados quantitativos, como mapas de distância, altitude e declividade). Uma questão fundamental para qualquer processo de classificação consiste na preservação da estrutura dos dados. O presente método, adaptado de Agterberg & Bonham-Carter (1990), calcula as faixas de acordo com a estrutura dos dados estabelecendo, primeiro, um delta mínimo – especificado como o incremento na interface gráfica – (Dx) para uma variável x de grandeza contínua que é usada para construir n buffers incrementais (Nx) compreendendo intervalos de $x_{minimum}$ para $x_{minimum} + nDx$. Cada n define um limiar que divide o mapa em duas classes: (Nx) e (\overline{Nx}). An é o número de células para um buffer (Nx) múltiplo de n e dn é o número de ocorrências para o evento modelado (D) dentro do buffer. As quantidades An e dn são obtidas para uma sequência ordenada de buffers $N(x_{minimum} + nDx)$. Logo após, os valores de W^+ para cada buffer são calculados usando as equações 2 a 4.

Uma sequência de valores An é plotada contra $An * \exp(W^+)$. Em seguida, subdivisões são determinadas para este gráfico aplicando um algoritmo de generalização de linha (Intergraph, 1991) que contém três parâmetros: 1) intervalo de distância mínima ao longo de x , $mindx$, 2) intervalo de distância máxima ao longo de x , $maxdx$, 3) ângulo de tolerância ft . Para dx (uma distância entre dois pontos ao longo de x) entre $mindx$ e $maxdx$, uma nova subdivisão é colocada sempre que $dx \geq maxdx$ (um ângulo entre v e v' - vetores que ligam o ponto atual ao último ponto e o último ao seu antecessor, respectivamente) excede o ângulo de tolerância ft . Portanto, o número de faixas decresce em função de ft . As faixas são, por fim, definidas ligando os pontos das subdivisões com linhas retas. Observe que An é praticamente livre de erro, enquanto dn está sujeito a uma considerável incerteza porque é tido como a realização de uma variável randômica. Uma vez que um pequeno valor de An pode gerar valores ruidosos para W^+ , Goodacre et al. (1993) sugere que, em vez de calculá-lo empregando as equações 2 e 4, se deveria estimar W^+ para cada faixa definida através da seguinte expressão:

$$W^+ = \ln\left(\frac{y_{n=k} - y_{n=k-1}}{A_{n=k} - A_{n=k-1}}\right) \quad (5)$$

onde $y_n = An * \exp(W^+)$ e k representa os pontos de inflexão definidos para os n incrementos de Dx (Fig.11).

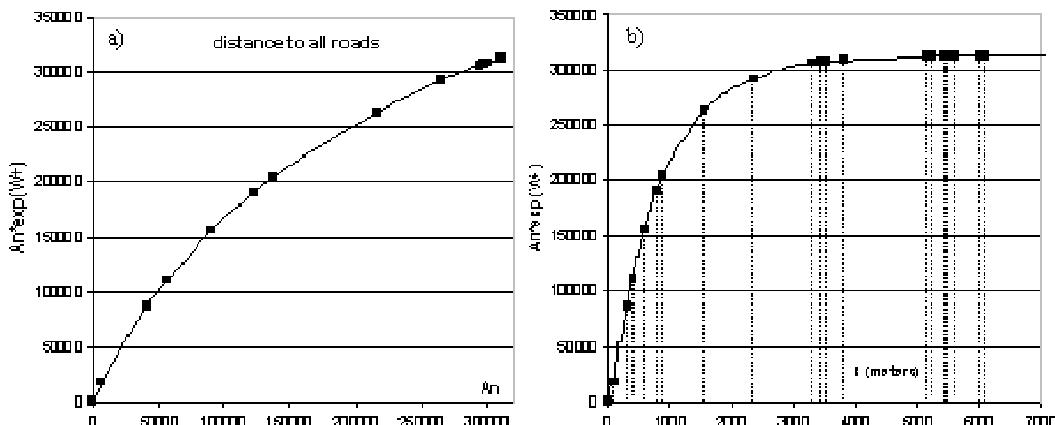
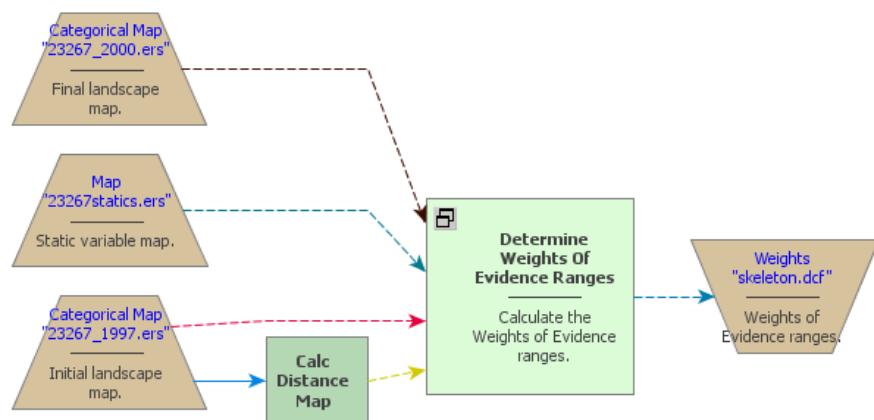


Fig. 11 a) Gráfico de An pela variável "distância até as estradas".

A curva de melhor ajuste pode ser aproximada por uma série de segmentos de retas usando um algoritmo de generalização de linhas como explicado no texto. Esta abordagem é usada para definir os pontos de inflexão para esta curva e, em seguida, intervalos de categorias para uma variável contínua (b).

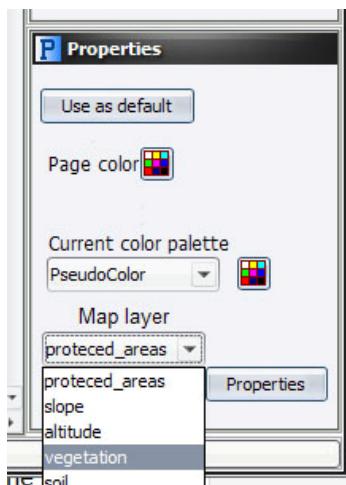
Agora, abra o modelo “determine_weights_of_evidence_ranges.xml” localizado em “\setup_run_and_validade_a_lucc_model\2_weights_of_evidence_ranges_calculation”.

Este modelo calcula faixas para classificar variáveis expressas em escalas contínuas de cinza para a derivação dos Pesos de Evidência. Ele seleciona o número de intervalos e o tamanho de seus buffers tendo como objetivo preservar a estrutura dos dados. Vá ao Help para uma descrição adicional a respeito deste método. Como resultado, sua saída é usada como entrada para o cálculo dos coeficientes dos Pesos de Evidência.

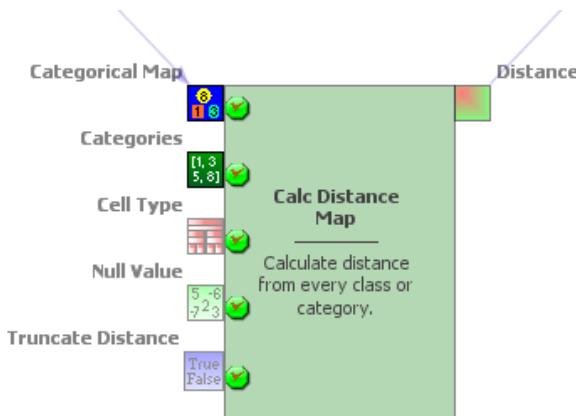


Em adição aos mapas de paisagem final e inicial, este modelo recebe um cubo de dados raster composto por uma série de mapas estáticos, e.g. vegetação, solo, altitude (eles são nomeados desta forma porque eles não se alteram durante a iteração do modelo. Um cubo raster inclui um conjunto de camadas de mapas co-registrados.

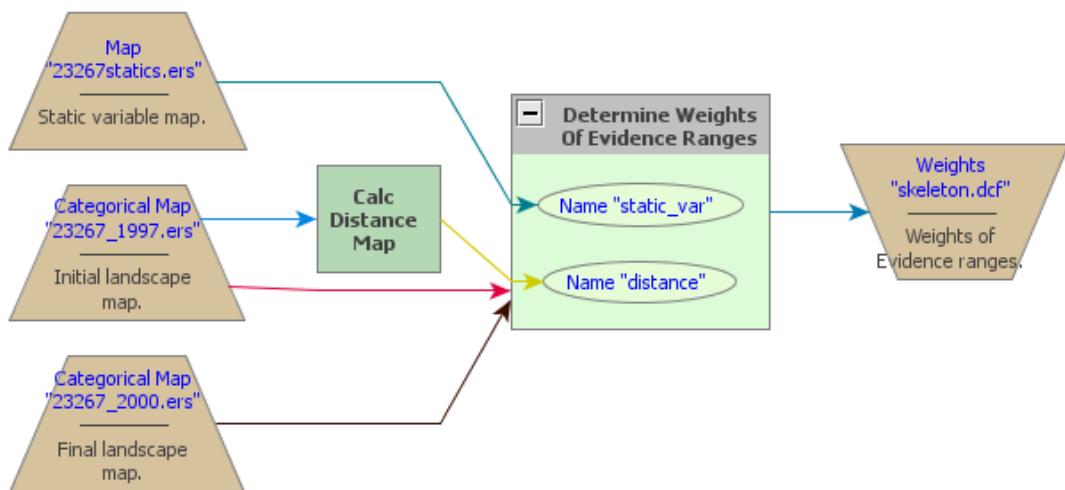
Abra o arquivo “23267static.ers” a partir da pasta “\setup_run_and_validade_a_lucc_model\originals” no Map Viewer. Uma opção para selecionar a camada irá aparecer na parte inferior esquerda do Map Viewer. Mude a camada para examinar os demais mapas. **DICA:** você pode construir um cubo raster reunindo um conjunto de mapas raster co-registrados através do functor *Create Cube Map* e extrair uma camada de um cubo raster usando o functor *Extract Map Layer*. Dados em cubo raster são suportados apenas no formato ER.



Além disso, Dinamica EGO pode incorporar camadas dinâmicas na simulação, as quais são assim conhecidas por poderem ser atualizadas durante a iteração do modelo. Para este modelo você irá incluir a variável “distância até áreas previamente desmatadas” como um mapa dinâmico. Para esta finalidade, o modelo emprega o functor *Calc to Distance Map*. Abra-o com o Edit Functor Ports.



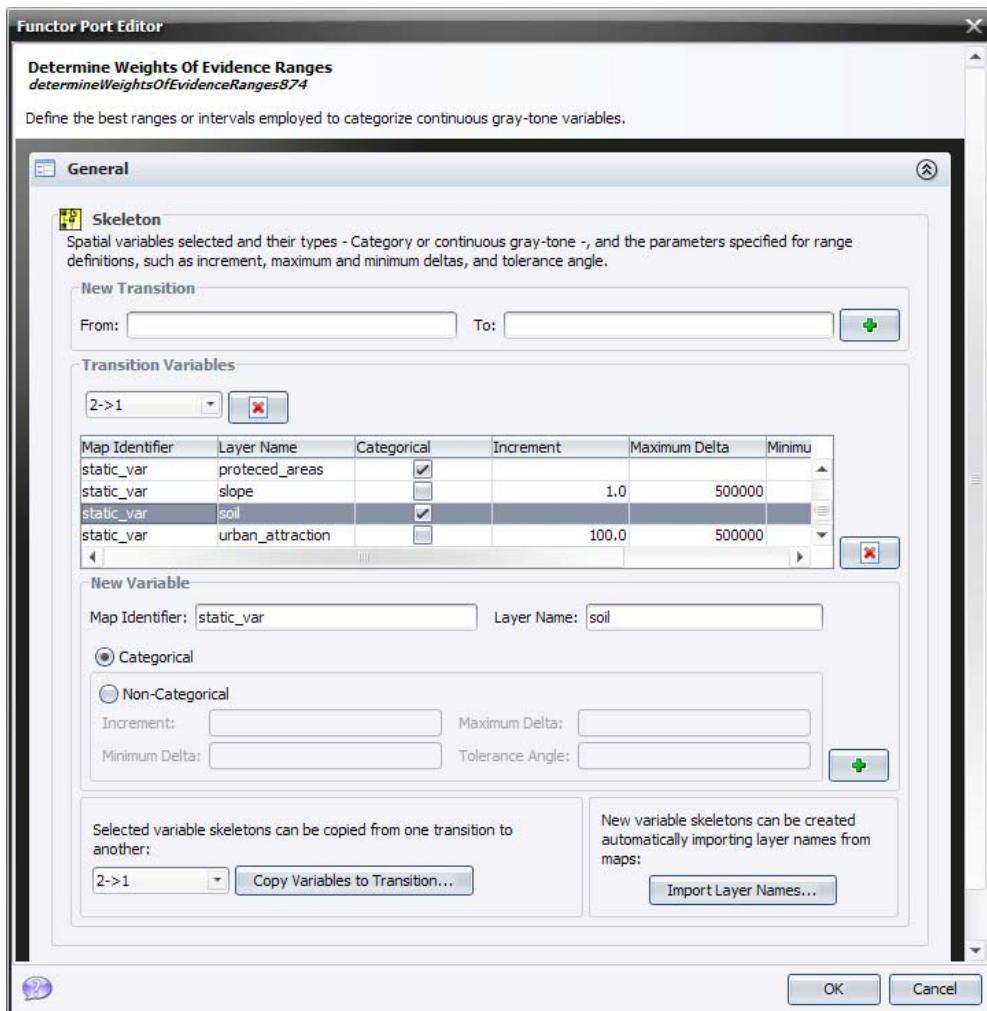
Este functor recebe como entrada um mapa categórico que, neste caso, é um mapa de paisagem. Clique agora no porto **Categories**. Este functor gera um mapa de distância de fronteira (distância mais próxima) entre células de cada classe de mapa definida pelo usuário. Neste caso, a classe “1” representa área desmatada. Assim, o modelo leva em consideração a proximidade de áreas previamente desmatadas em relação à probabilidade de novo desmatamento. Agora, abra o container *Determine Weights Of Evidence Ranges* clicando em seu ícone na parte superior esquerda.



No lugar de *Number Map*, agora você encontrará um *Name Map* dentro deste container. Esse functor é aplicado a containers que necessitam de um nome para o mapa ou um alias para identificar os mapas passados a eles. *Name Map* é encontrado na aba Map Algebra Supplementary. Pode ser qualquer nome, mas os mesmos nomes serão usados durante o ajuste de parâmetros internos do container, como é mostrado abaixo. Exemplos de containers que precisam de um *Name Map* são: *Determine Weights Of Evidence Ranges*, *Determine Weights Of Evidence Coefficients*, e *Calc WOfE Probability Map*.

Há dois functores *Name Map* dentro desse container, um para o mapa “23267statitcs.ers” e outro para o mapa de distância resultante do *Calc Distance Map*.

Agora abra *Determine Weights Of Evidence Ranges* com o Edit Functor. Ajuste o tamanho da janela como a seguir:



Observe que *Name Map* distância tem uma camada nomeada “distance_to_1” e static_var tem uma série de camadas, cada uma representando uma variável cartográfica.

DICA: Dos três formatos de arquivo suportados, somente o formato ER Mapper suporta nomes para camadas. Este é um formato BIL (Band Interleaved) com um arquivo ASCII separado para cabeçalho.

O cubo raster contém as camadas altitude, d_all_roads (d significa distância), d_major_rivers, d_paved_roads, d_settlement, d_trans_rivers, protected_areas, slope, urban_attraction (um mapa de potencial de interação), e vegetação.

DICA: Nomes de camadas devem ser os mesmos que os especificados no arquivo de cabeçalho. Dinamica EGO converte o formato Geotiff em formato ER Mapper, você precisa apenas conectar o *Load Map* ao *Save Map*.

Protected areas (áreas protegidas), vegetation (vegetação) e soil (solo) já são dados categóricos, portanto, marque-os como tais. Os outros precisarão ser ainda classificados. Os parâmetros para o processo de classificação são o incremento (incremento) – o incremento mínimo de buffer em unidades do mapa, por exemplo, em metros ou graus (**DICA:** no caso do mapa de distância, o incremento será equivalente à resolução da célula), os deltas máximos e mínimos que representam intervalos no eixo Y dos gráficos na fig. 11, e o ângulo de tolerância, o qual mede o ângulo de desvio a partir de uma linha reta. Alguns valores convencionais são

sugeridos. O resultado deste functor será um arquivo esqueleto dos Pesos de Evidência, mostrando os intervalos de categorização, porém, com todos os pesos iguais a zero. Execute o modelo e abra o arquivo resultante com um editor de texto. **DICA:** se o modelo tem mais de uma transição, você pode copiar e colar os parâmetros dos intervalos nas outras janelas de transições. Dinamica EGO permite a definição de diferentes faixas para cada transição.

```
:static_var/soil 5:9 9:10 10:11 11:13 13:14
2,1 0 0 0 0 0
:static_var/urban_attraction 0:100 100:200 200:1900
2,1 0 0 0
:static_var/vegetation 4:5 5:15 15:17 17:18
2,1 0 0 0 0
```

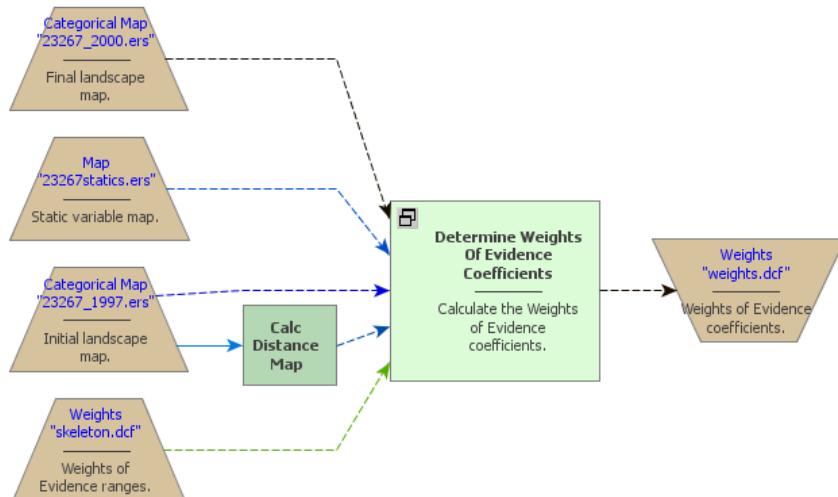
Exemplo de esqueleto para Pesos de Evidência

A primeira linha contém as faixas e a segunda a transição e seus respectivos coeficientes de Pesos de Evidência, os quais ainda são iguais a zero.

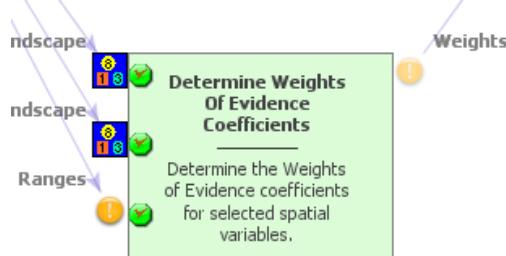
Embora os fundamentos matemáticos desse módulo sejam um pouco difíceis de compreender à primeira vista, ele provê um meio fácil de manipular modelos com multi-estados e transições. Vamos agora ao cálculo dos Pesos de Evidência.

7.3 Terceiro passo: Calculando os coeficientes dos Pesos de Evidência

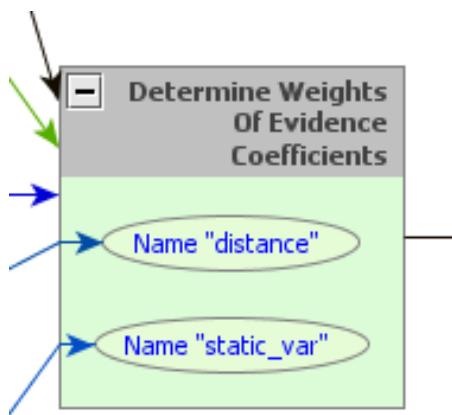
Abra o modelo “determine_weights_of_evidence_coefficients.xml” localizado em `setup_run_and_validate_a_lucc_model\3_weights_of_evidence_coefficient_calculation`.



O mesmo conjunto de dados da etapa anterior será empregado novamente, também o esqueleto da WEOF, o qual é carregado através do functor *Load Weights* localizado na aba input/output. Não há parâmetro a ajustar no functor *Determine Weights Of Evidence Coefficients*, você precisa apenas ajustar as conexões apropriadas, como é mostrado adiante:



Assim como os mapas de entrada:



DICA: Sempre use os mesmos nomes no *Name Map*.

Maximize a janela log e execute o modelo. Vamos analisar os resultados para a variável “distance_to_1”.

Range	Possible Transitions	Executed Transitions	Weight Coefficient	Contrast	Significant?
0 <= v < 500	61008	16874	1.42844	2.20576	yes
500 <= v < 750	40680	5181	0.465394	0.547235	yes
750 <= v < 1000	22527	1906	0.00859705	0.00922727	no
1000 <= v < 1250	14860	956	-0.287274	-0.299202	yes
1250 <= v < 1500	13224	682	-0.521908	-0.5393	yes
1500 <= v < 2000	14052	566	-0.780913	-0.805828	yes
2000 <= v < 2250	5719	175	-1.06579	-1.07792	yes
2250 <= v < 7750	66722	1220	-1.59333	-1.7918	yes
7750 <= v < 8000	2129	3	-4.17349	-4.18041	yes
8000 <= v < 41500	90089	224	-3.60452	-3.94792	yes

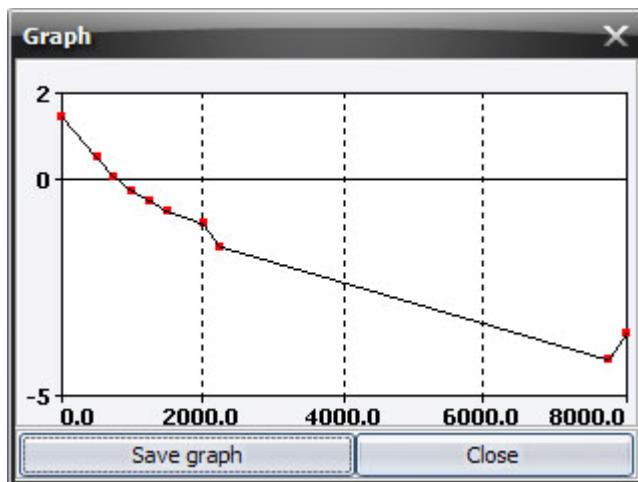
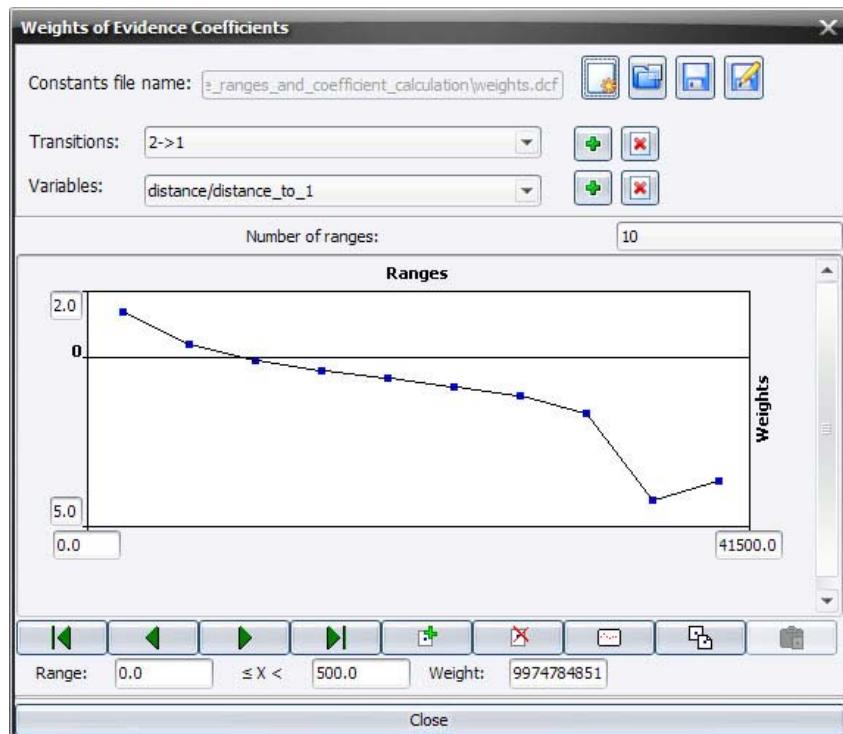
	331010	27787			

A primeira coluna mostra as faixas, a segunda o tamanho do buffer em células, a terceira o número de transições ocorridas dentro de cada buffer, a quarta os coeficientes obtidos, a quinta a medida do Contraste e a última o resultado para o teste de significância estatística. Para mais detalhes, vá ao Help. **DICA:** o Help do Dinamica EGO traz uma descrição detalhada de algoritmos mais complexos, tais como aqueles empregados nos functores de calibração e validação.

Observe que as primeiras faixas mostram uma associação positiva, favorecendo o desmatamento, especialmente a primeira. Em contrapartida, as faixas finais mostram valores negativos, portanto, repelindo o desmatamento. As faixas centrais mostram valores próximos de zero, significando que estas faixas de distâncias não exercem um efeito sobre o desmatamento.

O Contraste mede o efeito de associação/repulsão. Próximo de zero, não há efeito nenhum, enquanto que quanto maior e positivo o valor se torna, maior é a atração; por outro lado, quanto maior e negativo é o valor, maior é o efeito de repulsão.

Agora abra o editor gráfico dos Pesos de Evidência dentro do functor *Save Weights* (ícone do olho).



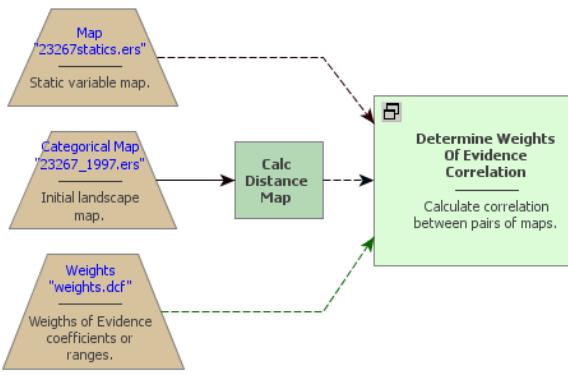
Você pode editar graficamente os coeficientes dos Pesos de Evidência e também pode obter uma visão de uma função contínua dos coeficientes dos Pesos de Evidência, clicando no botão Bird View. Observe agora como o desmatamento provável varia em função da distância até uma área previamente desmatada.

7.4 Quarto passo: Analisando a correlação de mapas

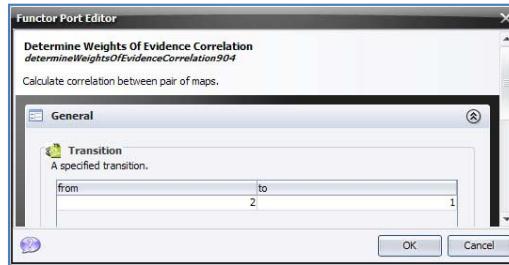
A única suposição necessária ao método dos Pesos de Evidência é que os mapas de entrada devem ser espacialmente independentes. Um grupo de medidas pode ser aplicado para estimar esta suposição, como o teste de Cramer e o Joint Information Uncertainty (Incerteza de Informação Conjunta) (Bonham-Carter, 1994). Como resultado, variáveis correlacionadas devem ser desprezadas ou combinadas em uma terceira que substituirá o par correlacionado no modelo.

Abra o modelo “weights_of_evidence_correlation.xml” da pasta `setup_run_and_validate_a_lucc_model\4_weights_of_evidence_correlation`.

O modelo em `3_and_2_weights_of_evidence_ranges_and_coefficient_calculation` corresponde aos passos 2 e 3 combinados.



mostrado abaixo:



Antes de executar o modelo, maximize a janela log. Esta é uma parte do log reportado

First Variable	Second	Cramer			Entropy			
		Chi^2	Crammer*	Contingency	Entropy	Uncertainty	Joint	Joint Information^n
distance/distance_to_1	static_var/altitude	307131	0.236164	0.578103	2.74436	0.187539		
distance/distance_to_1	static_var/d_all_roads	415843	0.370451	0.637919	2.46035	0.266327		
distance/distance_to_1	static_var/d_major_rivers	115816	0.154568	0.388945	2.88692	0.0652104		
distance/distance_to_1	static_var/d_paved_roads	331553	0.246373	0.592822	3.186	0.179856		
distance/distance_to_1	static_var/d_settlement	154589	0.290203	0.449104	2.26246	0.0506741		
distance/distance_to_1	static_var/d_trans_rivers	111301	0.150738	0.392312	2.9126	0.0622871		
distance/distance_to_1	static_var/protected_areas	443365	0.425621	0.648197	2.12443	0.313912		
distance/distance_to_1	static_var/slope	213313	0.156816	0.508435	2.82553	0.121546		
distance/distance_to_1	static_var/soil	252705	0.321329	0.540639	2.71575	0.155706		
distance/distance_to_1	static_var/urban_attraction	7616.25	0.0557844	0.110881	1.63899	0.0110797		
distance/distance_to_1	static_var/vegetation	95351.7	0.227917	0.367188	2.03765	0.06158		
static_var/altitude	static_var/d_all_roads	255518	0.289	0.542757	2.55059	0.178393		

DICA: Seria interessante copiar o log, colar em um editor de texto e depois exportar para uma planilha. Na planilha, você poderá ajustar uma tabela de cruzamento mostrando todos os pares de variáveis.

Vamos checar os pares correlacionados de mapas. O seguinte par chama a nossa atenção:

	Chi^2	Crammer*	Contingency	Entropy	Uncertainty	Joint I.
d_major_rivers -> d_trans_rivers	2.80782e+06	0.958014	0.906132	1.62982	0.885453	

Embora não haja concordância a respeito de qual limiar deve ser usado para excluir uma variável, todos os testes destacam uma alta correlação neste par de variáveis. Dessa forma, você deve excluir um dos dois. Vamos remover o d_major river. Apague a variável do arquivo dos Pesos de Evidências usando o editor gráfico. Abra-o no *Load Weights* usando o ícone do olho.

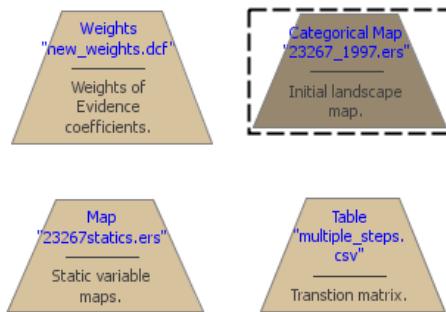
Agora salve os coeficientes WOFE como “new_weights.dcf” na pasta
`setup_run_and_validate_a_lucc_model\4_weights_of_evidence_correlation

Ótimo, você concluiu o processo de calibração, agora você pode começar a montar o modelo de simulação. Vamos adiante.

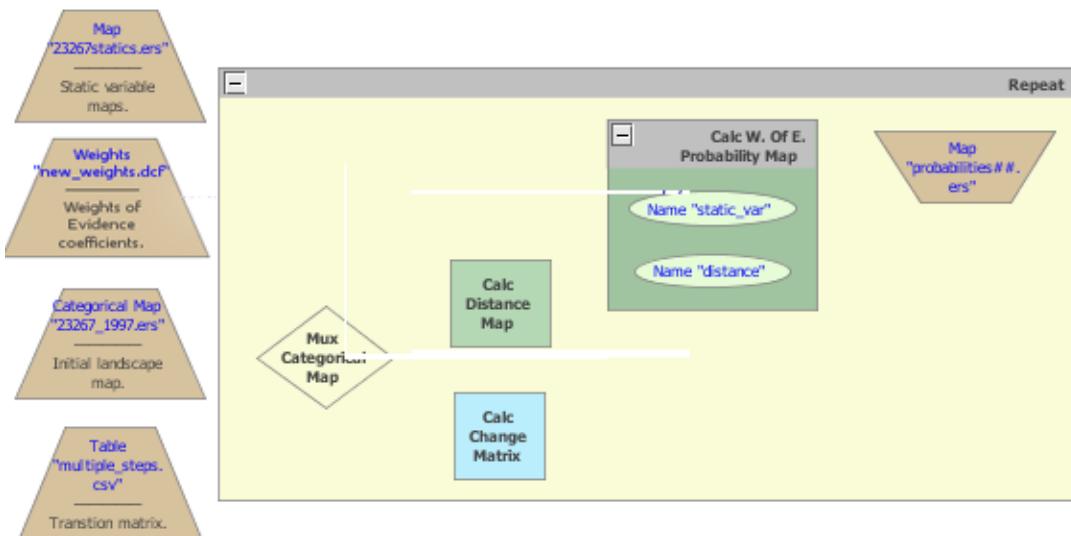
Este modelo executa testes em pares de mapas categóricos para testar a suposição de independência. Os métodos empregados são o Chi^2, Crammers, Contingência, a Entropia e o Joint Information Uncertainty (Incerteza de Informação Conjunta) (Bonham-Carter, 1994). Além das conexões a serem feitas, o único parâmetro a ser ajustado no functor *Determine Weights of Evidence Correlation* é a transição, como

7.5 Quinto passo: Ajustando e rodando um modelo de simulação LUCC

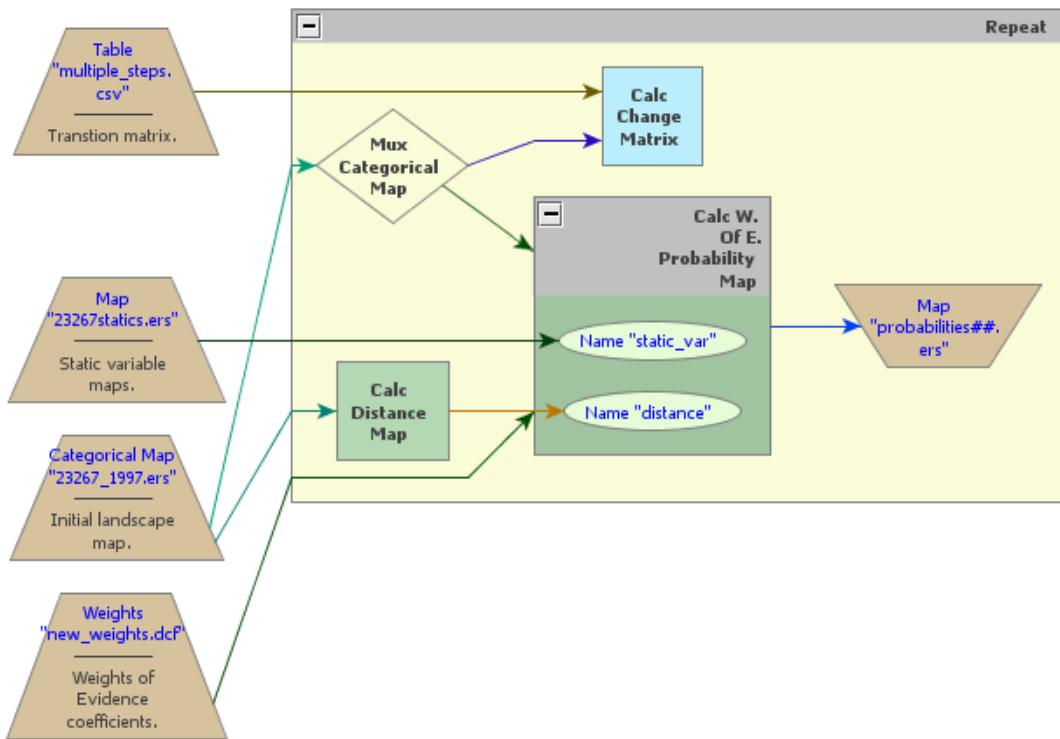
Vamos começar a ajustar o modelo de simulação do desmatamento carregando seus dados de entrada. Você precisará de um *Load Categorical Map* para carregar a paisagem inicial: “/original/23267_1997.ers”, um *Load Map* para “/originals/23267statics.ers”, um *Load Weights* para “new_weights.dcf”, e um *Load Lookup Table* para a matriz de transição de múltiplos passos: “multiple_steps.csv” porque você irá rodar o modelo em passos de tempo anuais. Adicione os seguintes comentários a cada functor:



Agora, insira um *Repeat*, e coloque dentro dele um *Calc Distance Map*, um *Mux Categorical Map* da aba Control, um *Calc Weights of Evidence Probability Map* e um *Save Map*. Coloque também dois functores *Name Map* dentro do *Calc Weights of Evidence Probability Map*. Renomeie-os com o nome utilizado no segundo e no terceiro passo. Abra o *Save Map* e escreva “probabilities.ers” na pasta \5_run_lucc. Deixe a opção **Suffix = “2”**.

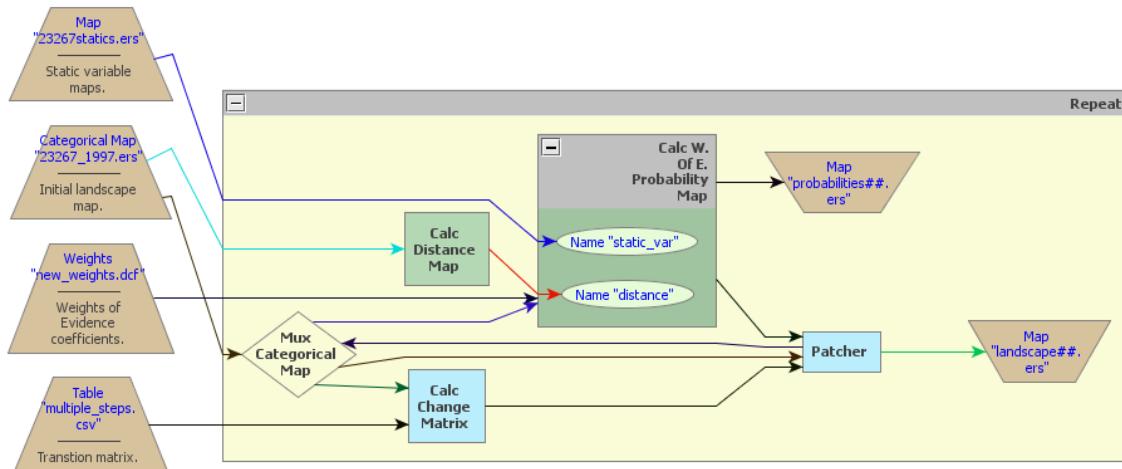


Vamos conectar os functores: primeiro, o mapa categórico “23267_1997.ers” ao porto **initial** do *Mux Categorical Map*, a saída deste ao *Calc Distance Map*, *Calc Change Matrix* e *Calc W. Of E. Probability Map*. Conecte a tabela “multiple_steps.csv” a *Calc Change Matrix*, o mapa “23267statics.ers” a nome “static_var”, a saída do *Calc Distance Map* ao nome “distance”, os pesos “new_weights.dcf” ao *Calc W. Of E. Probability Map* e sua saída ao mapa “probabilities##.ers” (Observe que o sufixo ## irá receber o passo da iteração do modelo).



Agora insira dois functores adicionais, *Patcher* da aba Simulation e outro *Save Map*. Escreva “Landscape.ers” e deixe “2” como **Suffix**. Conecte a saída do *Mux Categorical Map* ao porto do *Patcher* chamada **Landscape**, a saída do *Calc W. Of E. Probability Map* ao porto **Probabilities** do *Patcher*, e a saída do *Calc Changes* ao porto **Changes** do *Patcher*.

Para completar o loop você precisará conectar o porto de saída **Changed Landscape** do *Patcher* ao porto **Feedback** do *Mux Categorical Map* e ao mapa “Landscape##.ers” para salvar os mapas da execução do modelo. O modelo deverá, agora, se parecer com o seguinte:



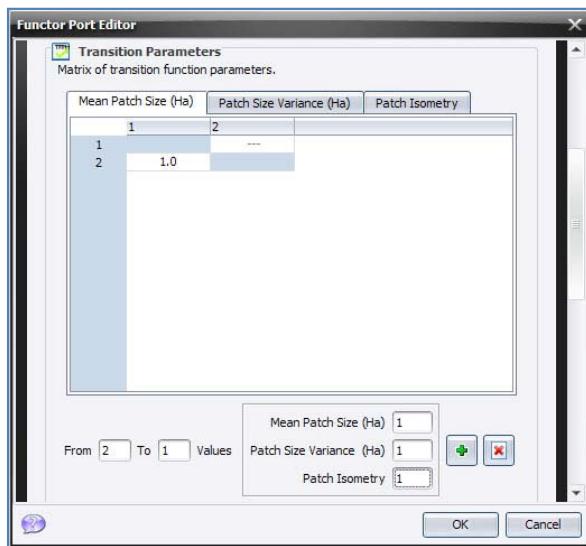
O functor *Mux Categorical Map* permite uma atualização dinâmica do mapa de paisagem de entrada. Ele recebe o mapa categórico “23267_1997.ers” em seu porto **Initial** no começo da simulação e, em seguida, o mapa de saída do *Patcher* através do porto **Feedback**.

O functor *Calc W. OF. E. Probability Map* calcula um mapa de probabilidade de transição para cada transição especificada somando os Pesos de Evidência, utilizando a equação 4.

Por sua vez, o *Calc Change Matrix* recebe a matriz de transição, composta por taxas líquidas, e a utiliza para calcular taxas brutas em termos de quantidade de células a serem alteradas multiplicando as taxas de transição pelo número de células disponíveis para uma mudança específica.

Dinamica EGO usa como regra local de Autômato Celular um mecanismo de transição composto por dois functores complementares de transição, *Expander* e *Patcher*, especialmente desenvolvidos para reproduzir os padrões espaciais de mudança (ambos são encontrados na aba Simulation). O primeiro processo é dedicado apenas à expansão ou contração de manchas já existentes de certa classe, enquanto o segundo processo foi feito para gerar ou formar novas manchas através de um mecanismo de semeadura. O *Patcher* procura por células ao redor de uma localização escolhida para uma transição conjunta. O processo é iniciado selecionando a célula núcleo da nova mancha e então selecionando um número específico de células ao redor da célula núcleo, de acordo com suas probabilidades de transição P_{ij} .

Variando seus parâmetros de entrada, estas funções possibilitam a formação de uma variedade de tamanhos e formas de manchas de mudanças. Patch **Isometry** varia de 0 a 2. As manchas assumem uma forma mais isométrica à medida que este número aumenta. Os tamanhos das manchas de alteração são ajustados de acordo com uma distribuição de probabilidade lognormal. Portanto, é necessário especificar os parâmetros desta distribuição representados pela média e a variância dos tamanhos das manchas a serem formadas. Abra este functor com Edit Functor e insira seus parâmetros conforme o que se segue (deixe os demais inalterados):

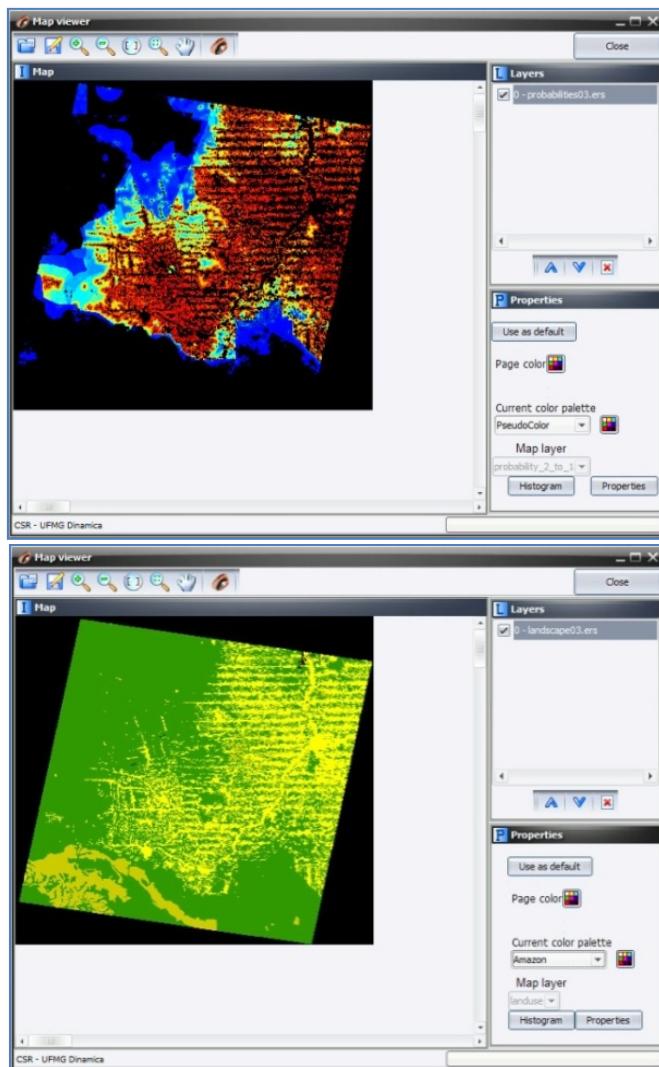


Uma vez que a resolução do mapa de entrada é aproximadamente de 250 metros, o ajuste do **Mean Patch** e de **Patch Size Variance** para “1” não permitirá a formação das manchas.

Agora vamos ajustar os parâmetros dos demais functores. Abra *Calc Distance Map* e insira “1”. Lembre-se que você quer “distance_to_1” (áreas desmatadas). Abra *Calc W. Of E. Probability Map* e insira a transição “2 para 1”, que representa desmatamento. Por fim, abra *Repeat* e ajuste **Number of Iterations** para “3”, pois você irá rodar o modelo em passos de tempo anuais.

Finalmente, verifique o modelo e execute-o. Verifique o Log reportado e abra os mapas “probabilities3.ers” usando “PseudoColor” como **Color Palette** e “landscape3.ers” usando

“Amazon” como Color Palette no Map Viewer. **DICA:** Quando *Save Map* é colocado dentro de *Repeat*, ele salva um mapa como resultado a cada passo de tempo (Se você quiser somente a paisagem simulada final, arraste o *Save Map* para fora do *Repeat*).



Observe as áreas com alta probabilidade de desmatamento e compare o mapa simulado “landscape3.ers” com a paisagem final “23267_2000.ers”. Eles se parecem? Para fazer uma comparação quantitativa, vamos para o passo seguinte.

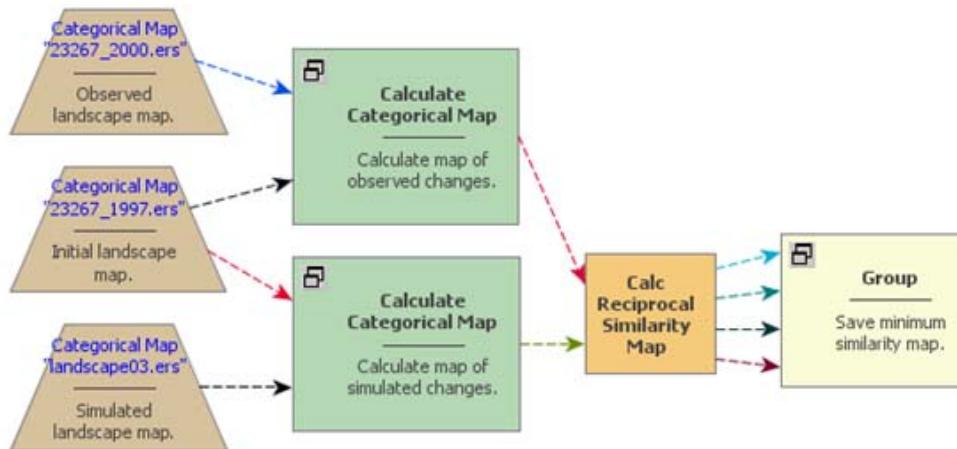
7.6 Sexto passo: Validando a simulação usando uma função de decaimento exponencial

Modelos espaciais requerem uma comparação em um contexto de vizinhança, porque até mesmo mapas que não se assemelham exatamente célula-a-célula podem ainda apresentar padrões similares e, igualmente, concordância espacial nas proximidades de uma célula. Para lidar com essa questão, muitos métodos de comparação baseados em proximidade têm sido desenvolvidos. Por exemplo, Costanza (1989) introduziu procedimento de ajuste de resoluções múltiplas que compara o ajuste de um mapa em tamanhos crescentes de janelas. Pontius (2002) apresentou um método similar ao de Costanza (1989) que diferencia erros devidos à localização e à quantidade. Power et al. (2001) forneceu um método de comparação baseado em casamento de padrão fuzzy hierárquico. Por sua vez, Hagen (2003) desenvolveu novas métricas, incluindo o Kfuzzy, considerado como equivalente à estatística Kappa, e a

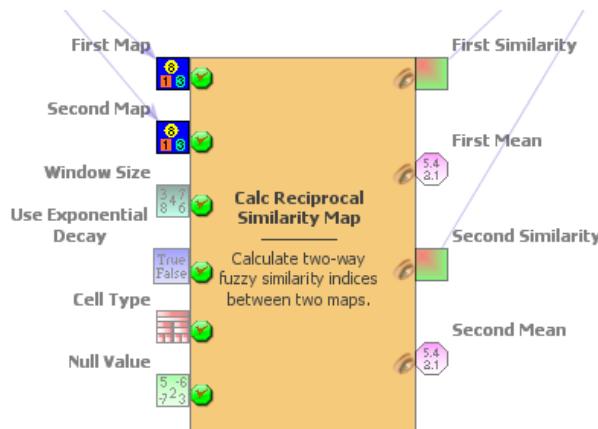
similaridade fuzzy que leva em consideração a incerteza da localização de uma categoria dentro da vizinhança de uma célula.

O método que aplicamos aqui é uma modificação deste último e é nomeado no Dinamica EGO como *Calc Reciprocal Similarity*. Este método emprega uma função de decaimento exponencial com a distância para ponderar a distribuição do estado da célula ao redor de uma célula central (Veja o esquema na fig.12 e depois vá ao Help para mais detalhes sobre este método).

Abra o modelo “determine-similarity-of-differences.xml” da pasta \6_validate_using_exponential_decay_function.



Como arquivos de entrada, o modelo recebe as paisagens final e inicial, e a paisagem final simulada. Observe que há dois funtores *Calculate Categorical Map* antes do *Calc Reciprocal Similarity Map*. Os mapas simulados herdam os padrões espaciais do mapa inicial de paisagem. Para remover esta herança, este modelo avalia o ajuste espacial entre mapas de mudanças.



Abra o *Calculate Categorical Map*, você irá encontrar a seguinte equação:

if (i1 = i2) then null else i2

Dessa maneira, o mapa resultante representará apenas as células que não mudaram. *Calc Reciprocal Similarity Map* calcula uma similaridade duas vias, do primeiro mapa para o segundo e do segundo para o primeiro. É aconselhável sempre escolher o menor valor de similaridade, uma vez que mapas randômicos tendem a produzir um alto ajuste artificial quando comparados univocamente, porque eles dispersam as mudanças sobre todo o mapa. Este teste emprega uma função de decaimento exponencial em uma janela de tamanho 11x11, sendo os valores truncados fora da janela.

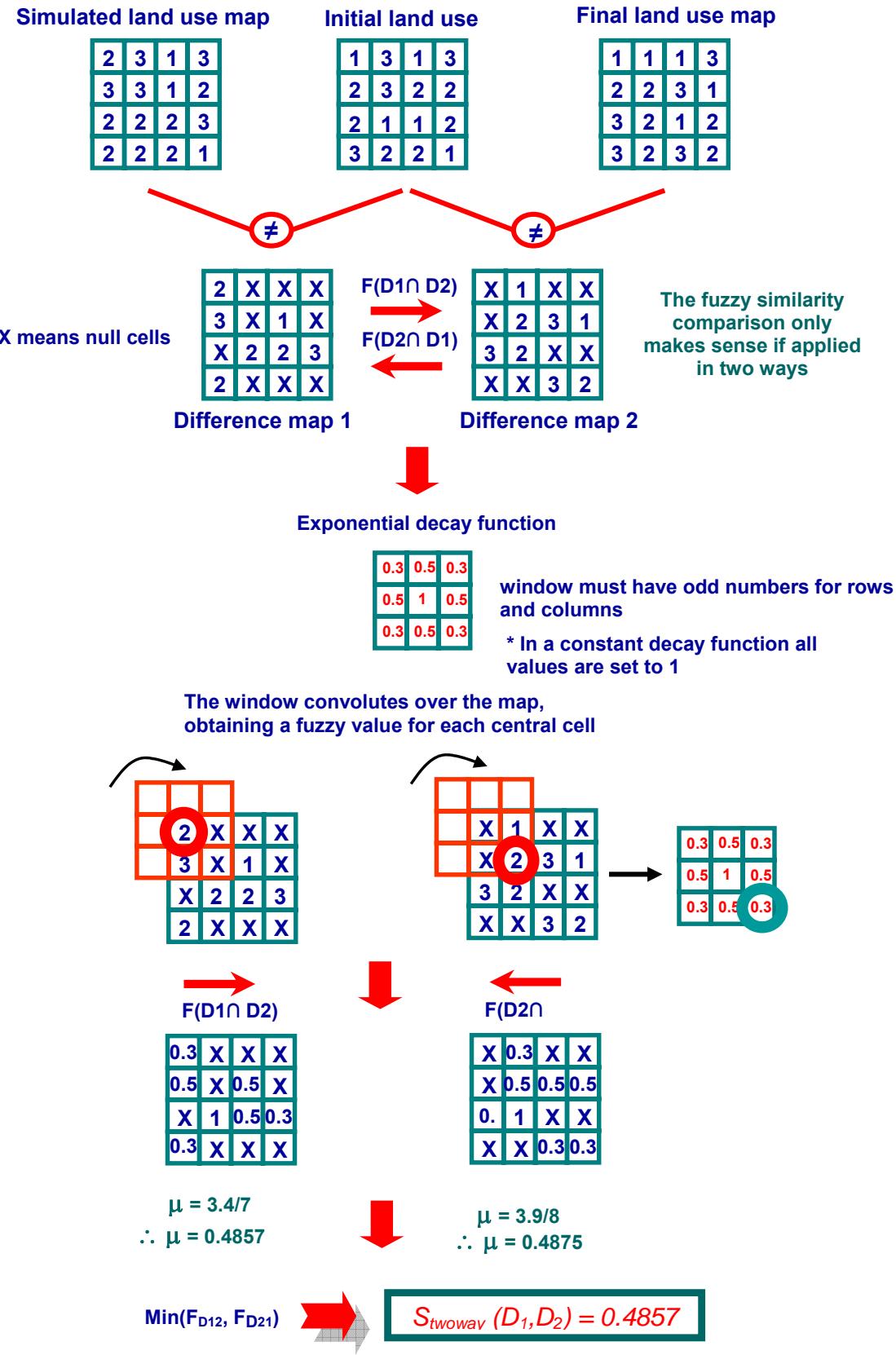
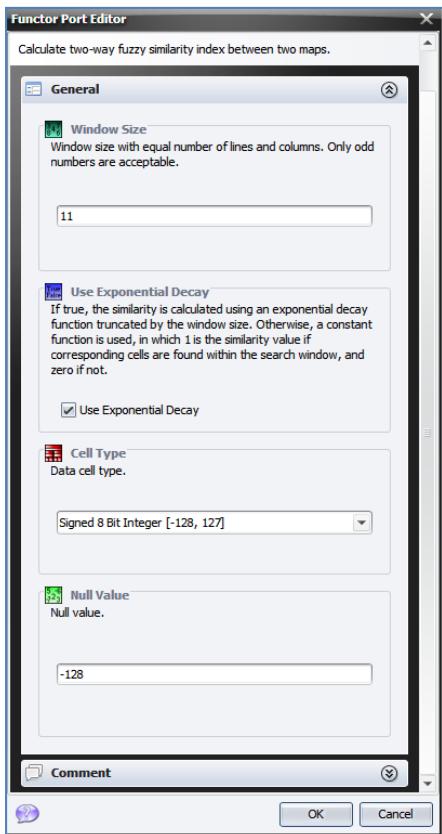


Fig. 12 – Método de comparação Fuzzy usando mapa de diferenças e uma função de decaimento exponencial. O mesmo processo se aplica a função de decaimento constante, na qual todos os pesos da janela são ajustados para 1.

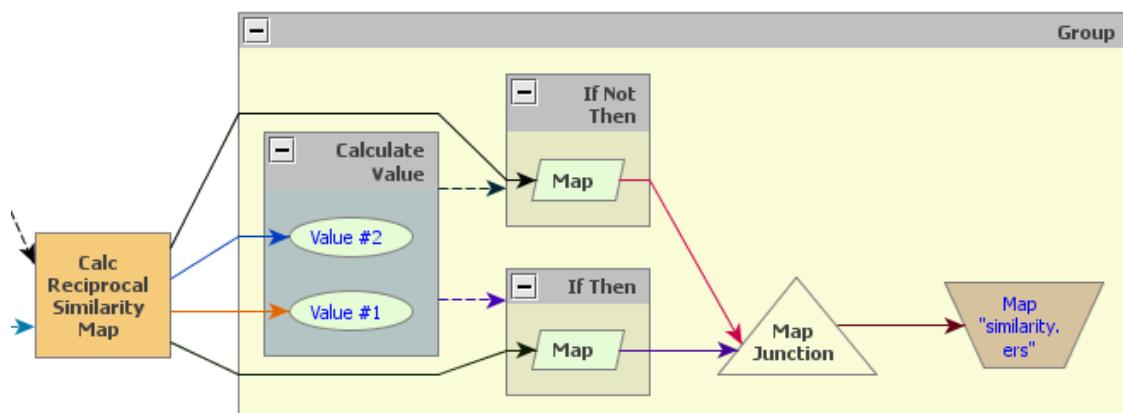


Abra *Calc Reciprocal Similarity Map* com o editor de portos:

Este functor recebe como entrada dois mapas, o primeiro e o segundo, e gera como resultado dois valores de similaridade, o **First Mean** e o **Second Mean**. Agora abra o functor com o Edit Functor.

Os dois parâmetros a serem ajustados são o **Window Size** e o **Use Exponential Decay**.

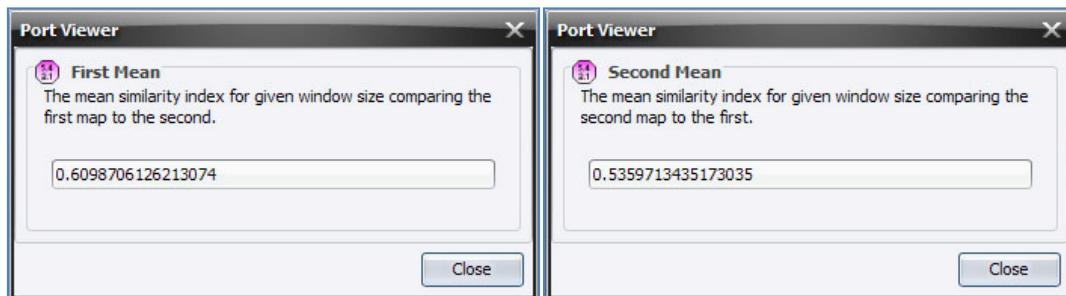
Abra o Group "Salva mapas de similaridade mínima".



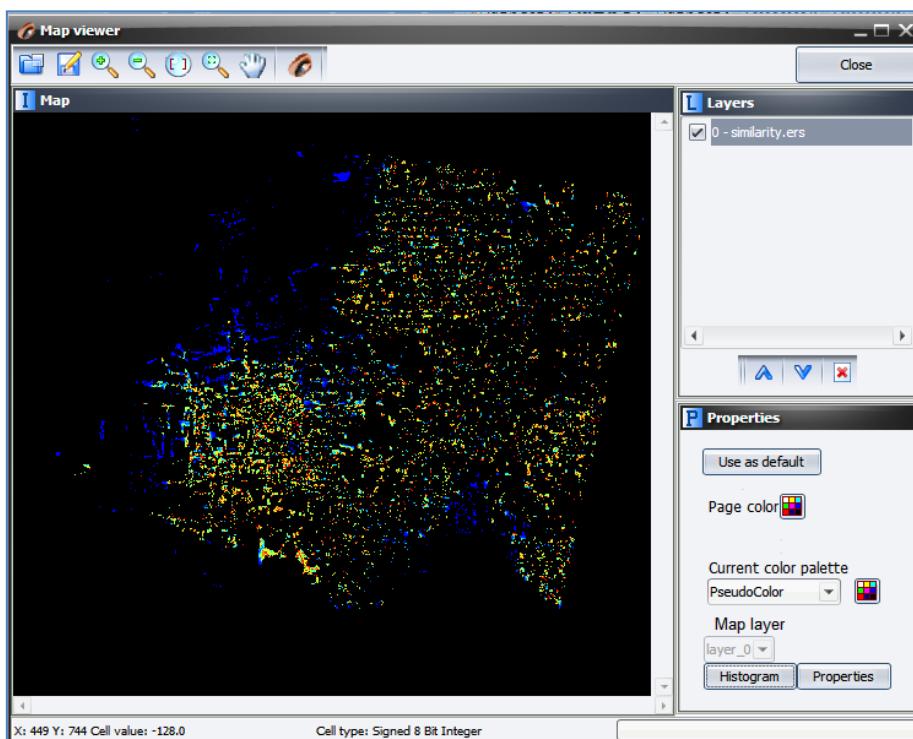
Este conjunto de functores permite a escolha do mapa de similaridade mínima a ser salvo. Para fazer isso, nós introduzimos três novos functores: *If Then*, *If Not Then* e *Join Map* – disponível na aba Control. Os dois primeiros são containers que recebem uma chave booleana como entrada (0 nega a condição e qualquer número maior que 0 toma a condição como verdadeira). Antes deles, o *Calculate Value* examina os dois valores de saída do *Calc Reciprocal Similarity Map*, o **First Mean** e o **Second Mean**, passando 0 ou 1 dependendo de qual dos dois é maior. *If Then* contém um *Calculate Map* que recebe o mapa **First Similarity** e o *If Not Then* que recebe o mapa **Second Similarity**. Dependendo do resultado booleano do *Calculate Value*, um dos dois containers irá passar a *Map Junction* o seu resultado, permitindo, assim, que o mapa com menor valor de similaridade seja salvo.

DICA: Estes três novos functores permitem o desenvolvimento de modelos que possuem bifurcações para dois ou mais fluxos de execução. Veja a fig. 1 no começo deste guia prático. Esta é uma característica fantástica para o desenvolvimento de modelos complexos.

Examine agora os portos **First Mean** e **Second Mean** clicando sobre elas com o botão direito.



O mapa de similaridade mínima corresponde à similaridade obtida comparando as mudanças simuladas com as mudanças reais. Visualize agora o mapa de similaridade resultante, abrindo-o no Map Viewer. Use **PseudoColor**, **Limits to Actual**, e **Histogram Equalize**.

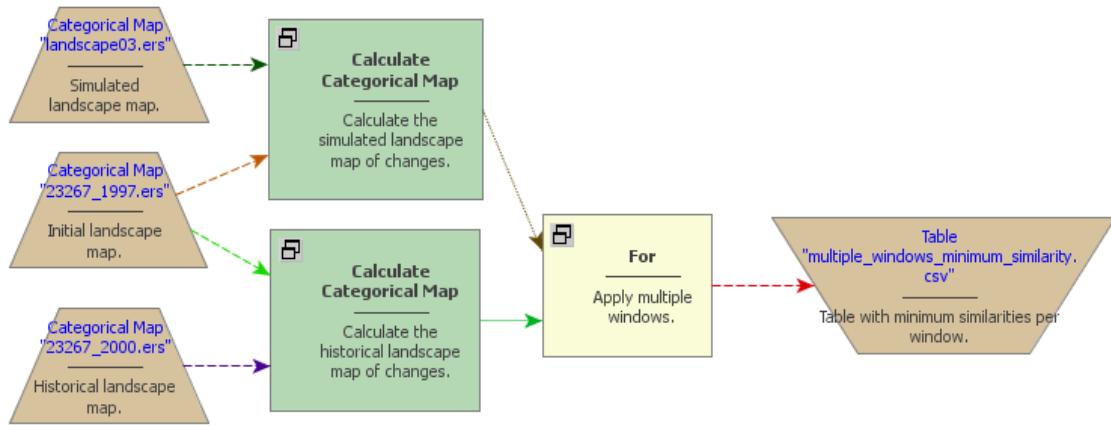


As áreas em vermelho e amarelo mostram altos a moderados ajustes espaciais, enquanto as áreas em azul indicam pouco ajuste.

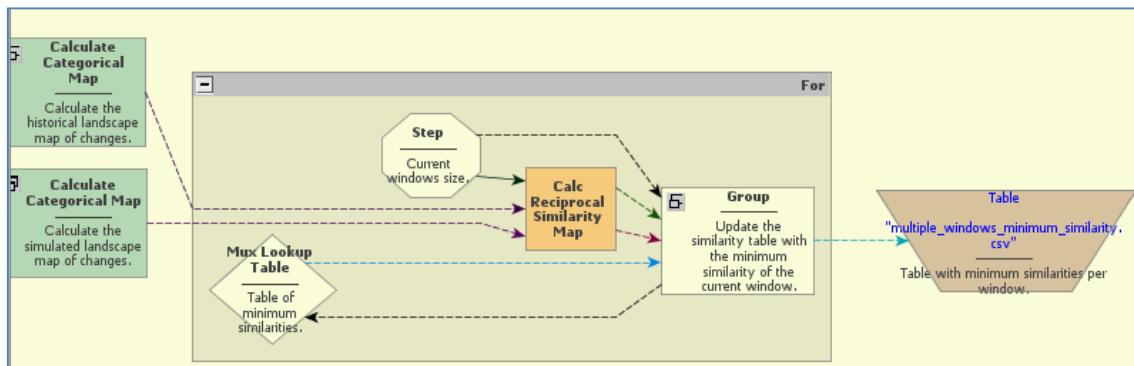
Outro jeito de medir o ajuste espacial entre dois mapas é por meio da análise de similaridade por múltiplas janelas. Este método emprega uma função de decaimento constante dentro de um tamanho variável de janela. Se o mesmo número de células de mudança é encontrado dentro da janela, o ajuste terá valor 1, não importando suas localizações. Isso representa um modo conveniente de avaliar o ajuste do modelo através de uma resolução espacial decrescente. Modelos que não se combinam bem em alta resolução talvez consigam um ajuste apropriado a uma resolução mais baixa. Vamos desenvolver uma comparação de ajuste em resoluções múltiplas no próximo passo.

7.7 Sétimo passo: Validando a simulação usando janelas múltiplas e função de decaimento exponencial

Abra o modelo “determine-multi-window-similarity-of-differences.xml” da pasta \7_validate_using_multiple_windows_constant_decay_function.

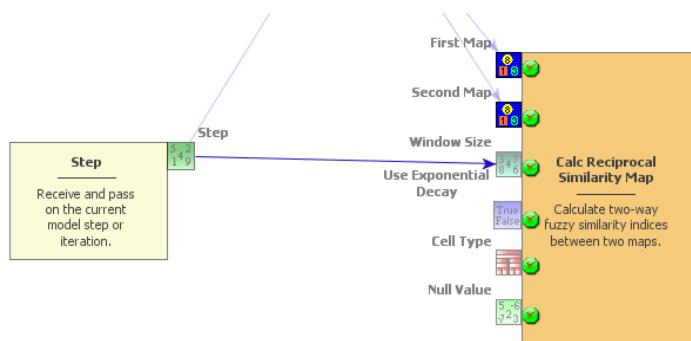


A primeira parte deste modelo é similar ao anterior. Dois functores *Calculate Map* são empregados para obtenção dos mapas de mudanças. Agora abra o functor *For*. Vamos examinar seu conteúdo em detalhe.



Ao centro temos um *Calc Reciprocal Similarity Map*. Abra-o com o *Edit Functor*. Observe que a opção **Use Exponential Decay** está inativa, o que significa que uma função de decaimento constante está sendo usada.

Clique na ligação entre esse functor e o *Step* com o *Edit Functor Port*.

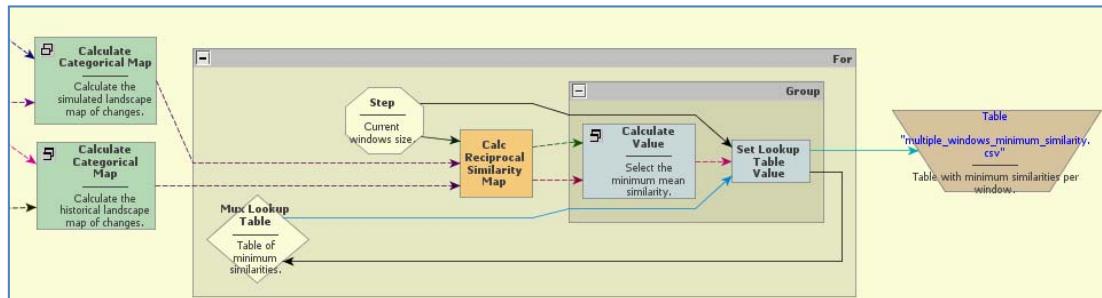


O Step vindo do For está controlando o **Window Size**. For é um caso particular de Repeat, no qual o passo inicial e o final, assim como o incremento de cada passo, podem ser definidos como se segue:

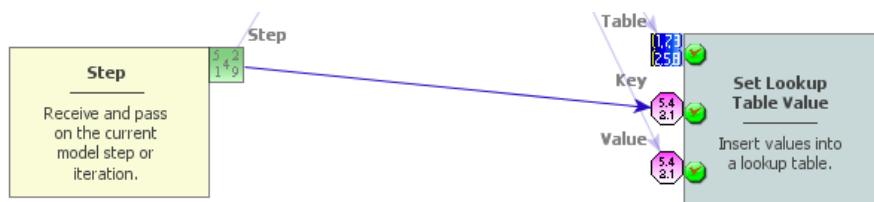


Neste container *For*, os passos vão de 1 a 11 por um incremento de dois. Isso é necessário porque o **Window Size** deve ser composto por números ímpares. Como resultado, **Window Size** irá variar de 1x1, para 3x3, 5x5, 7x7, 9x9, e 11x11.

Da mesma maneira que na lição 2, *Mux Lookup table* é empregado para atualizar a tabela que contém as médias das similaridades mínimas por tamanho de janela. Abra *Group* contendo os functores que atualizam estas tabelas.

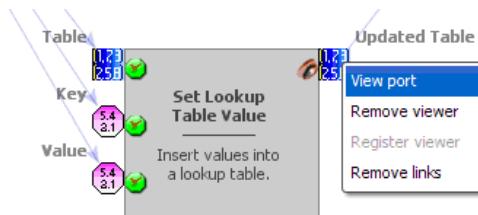


Calculate Value seleciona o menor valor de ajuste (a fórmula aplicada é $\min(v1, v2)$) e o envia ao *Set Lookup Table*, o qual também recebe como entrada o passo atual como a chave da tabela.

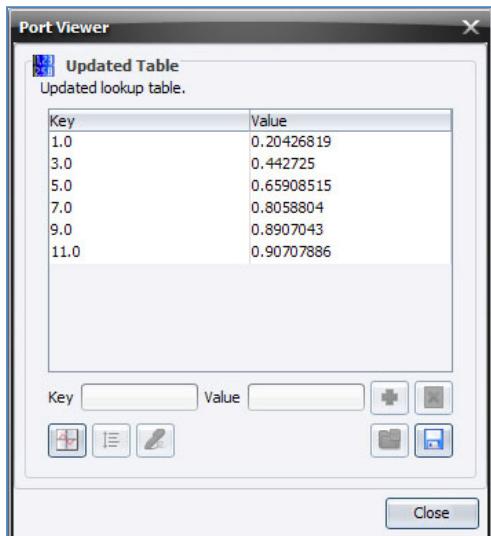


À medida que *For* itera, *Calc Reciprocal Map* calcula os valores de similaridade para um tamanho de janela e envia-os para *Calculate Value*, que seleciona o valor mínimo e o envia para *Set Lookup Table* que, por sua vez, atualiza o valor para a chave da tabela correspondente ao passo atual do modelo e retorna a tabela atualizada ao *Mux Lookup Table*. Quando *For* é completado, a tabela é repassada a *Save Table*.

Clique no Register viewer do porto **Updated Table** de *Set Lookup Table Value*, execute o modelo e analise a tabela resultante clicando no porto **Updated Table** com o botão direito.



Isto é o que você obterá.



O ajuste vai de 21% na resolução de 1x1 célula até 90% na resolução de 11x11 células. Observe que, o fato da simulação receber como entrada uma matriz de transição fixa para o ajuste da quantidade de mudanças, nós precisamos apenas avaliar o ajuste do modelo em relação à localização das mudanças. Levando em consideração que a resolução é de 250 metros e o raio de procura da janela tem a metade desta resolução, você pode desenhar um gráfico representando o ajuste do modelo por resolução espacial (fig. 13). **DICA:** abra a tabela com editor de gráficos.

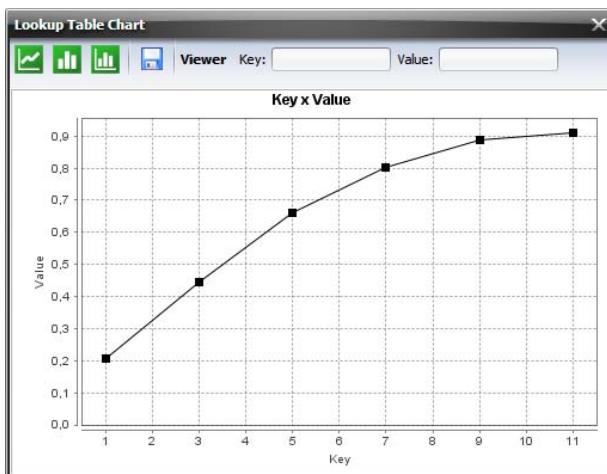


Fig. 13 – Ajuste do modelo

A partir do gráfico da fig. 13, podemos afirmar que a simulação atingiu um valor de ajuste de similaridade de até 50% em uma resolução espacial de ≈ 800 metros. **DICA:** Este método aplica a múltiplas transições também.

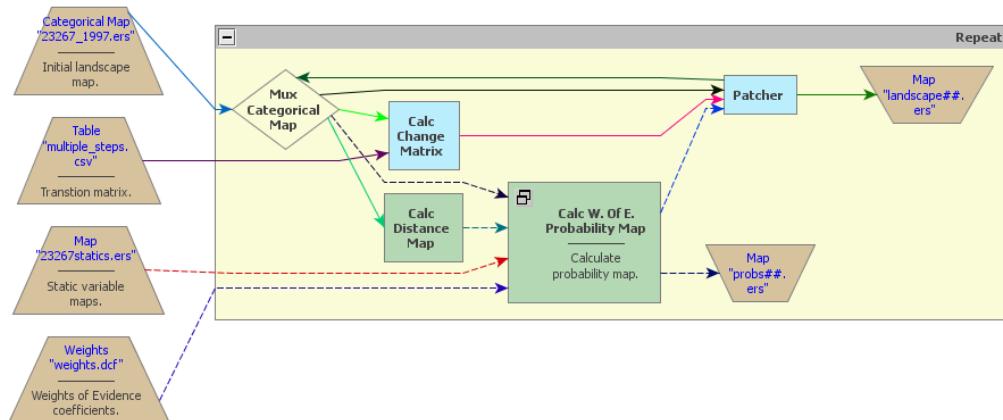
Agora que a fase de validação foi completada, você pode começar a brincar com os parâmetros do modelo de simulação para explorar possíveis resultados. Façamos isso no próximo passo.

7.8 Oitavo passo: Executando simulação com formação de manchas

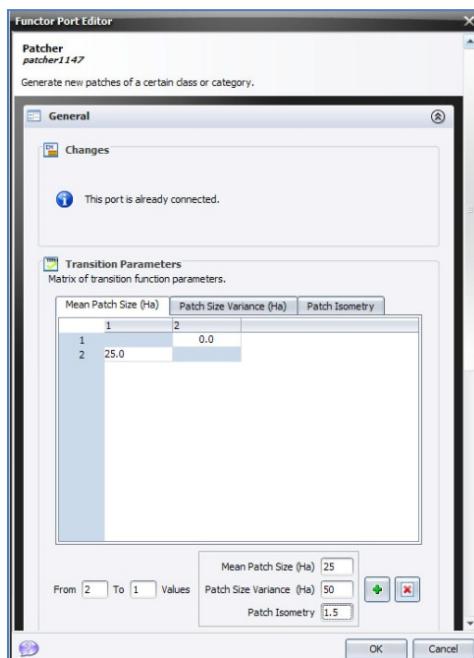
Antes de passar para esta etapa, abra e leia o arquivo "simulating_the_spatial_patterns_of_change.pdf" na pasta Examples\patterns_of_change. Este texto apresenta e discute os

resultados de uma série de simulações usando mapas sintéticos simplificados, variando os parâmetros das funções de transição. Os resultados da simulação são avaliados usando métricas de estruturas de paisagem, tais como dimensão fractal, índice de agregação de manchas e distância até o vizinho mais próximo. Estes exemplos de simulação são usados para mostrar como essas funções podem ser calibradas, assim como seu potencial para replicar os padrões espaciais mutantes de uma variedade de fenômenos dinâmicos (Soares-Filho *et al.*, 2003). Você pode também examinar os modelos disponíveis na pasta Examples\patterns_of_change.

Esta etapa tem por objetivo analisar os parâmetros da função de transição do Patcher na estrutura da paisagem simulada. Abra o modelo “simulate_deforestation_from_1997_2000_with_patch_formation” da pasta \Examples\setup_run_and_validate_a_lucc_model\8_run_lucc_with_patch_formation. Este é o mesmo modelo usado no quinto passo.

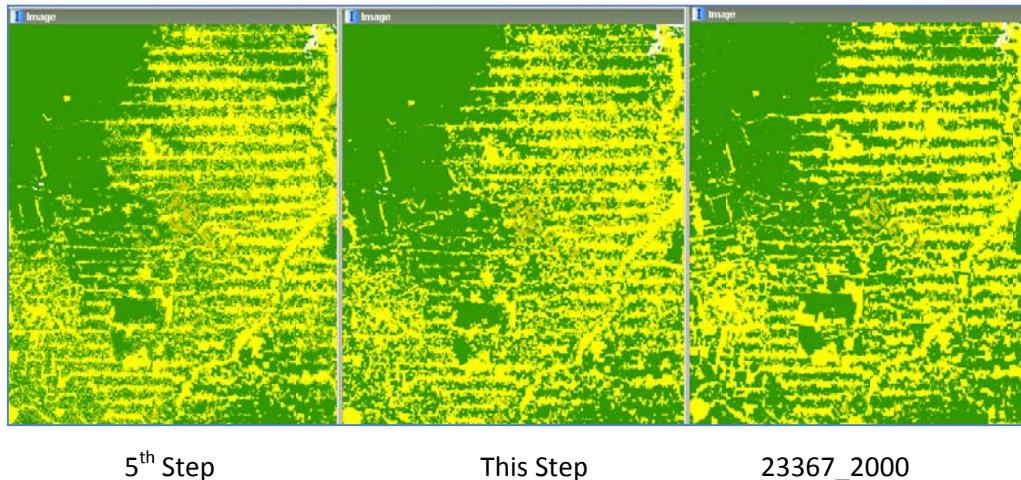


Agora abra o functor Patcher com a ferramenta Edit Functor.



Mean Patch Size está ajustado em 25 ha, **Patch Size Variance** em 50 ha, e o **Patch isometry** em 1.5. Uma vez que o tamanho da célula é igual a 6.25 hectares (250 x 250 metros), as manchas a serem formadas terão em média 4 células e uma variância de 8 células.

Execute o modelo, abra ambos os arquivos “landscape3.ers” deste modelo e do quinto passo, assim como o arquivo “23267_2000.ers” para comparação visual.



Observe que a paisagem simulada neste passo apresenta uma estrutura de paisagem próxima àquela da paisagem final observada. Vamos agora adicionar o functor *Expander* ao modelo de simulação.

7.9 Nono passo: Executando a simulação com formação de manchas e expansão

O functor *Expander* se dedica exclusivamente à expansão ou à contração de manchas existentes de certa classe. Portanto, no *Expander*, uma nova probabilidade espacial de transição P_{ij} depende da quantidade de células do tipo j ao redor de uma célula do tipo i , como representado na fig. 14.

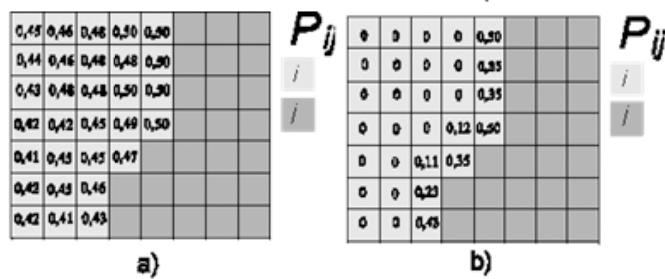
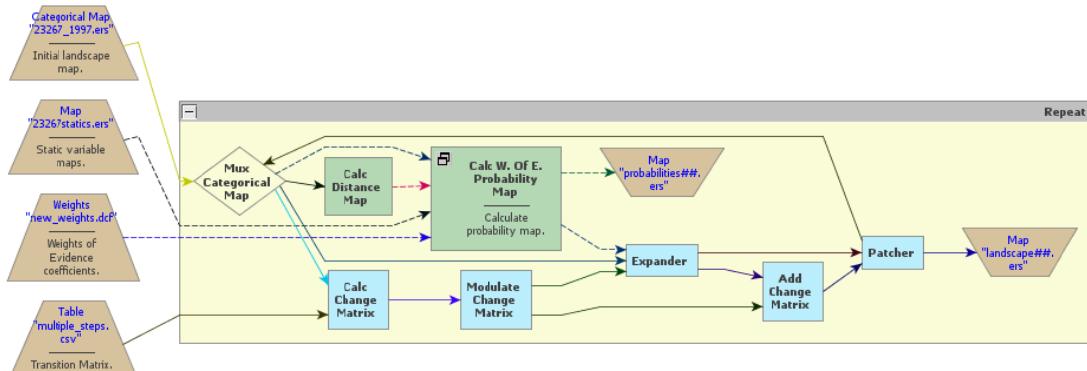
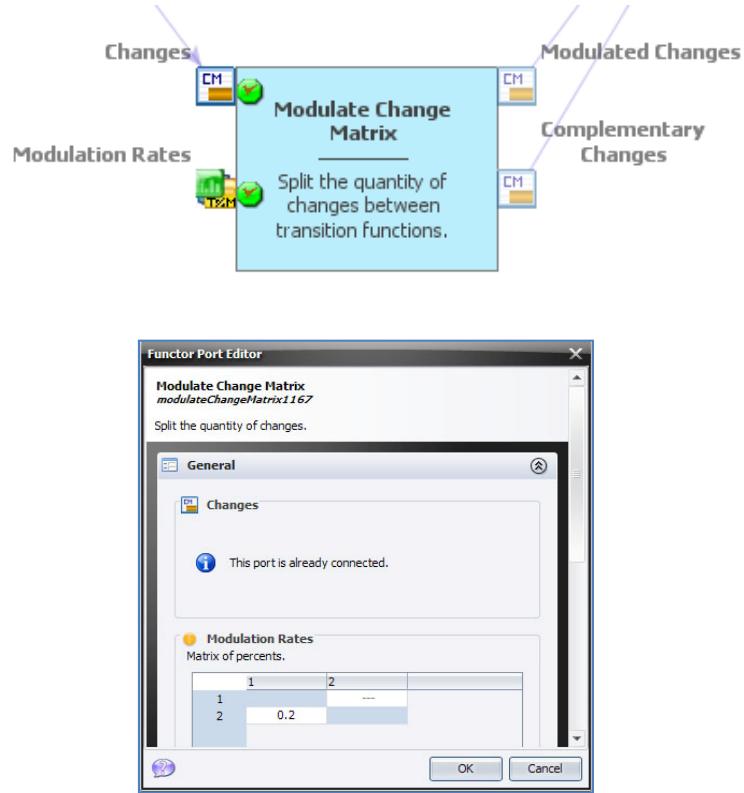


Fig. 14 – Probabilidades de transição transformadas para simular processo de expansão.

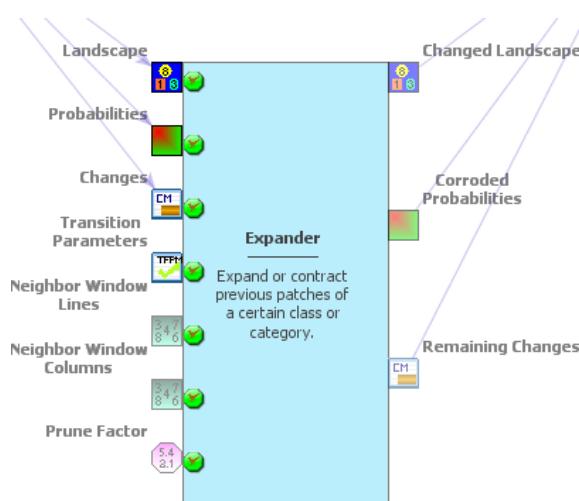
Agora, abra o modelo “simulate_deforestation_from_1997_2000_with_patchFormation_and_expansion.xml” na pasta \\Examples\\setup_run_and_validate_a_lucc_model\\9_run_lucc_with_patchFormation_and_expansion.



Este modelo difere do anterior pela presença de três novos functores: *Modulate Change Matrix*, *Expander*, e *Add Change Matrix*. *Modulate Change Matrix* divide o número de células a serem alteradas por transição em duas matrizes: a matriz **Modulated Changes** e a **Complementary Changes**. A primeira vai para *Expander* e a segunda para *Add Change Matrix*. Neste caso, 20% das mudanças 2 para 1 vão para *Expander*.

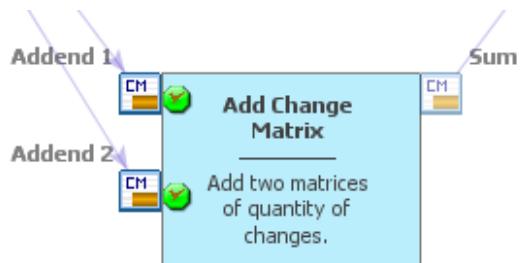


Além disso, o mapa de probabilidade vindo do *Calc W.OF E. Probability Map* vai primeiro a *Expander*, normalmente colocado antes do *Patcher*, porque não é possível garantir que *Expander* irá executar a quantidade total de mudanças passadas a ele. Você precisará, ainda, conectar seus portos de saída **Changed Landscape** e **Corroded Probabilities** a *Patcher*, e **Remaining Changes** a *Add Change Matrix*.

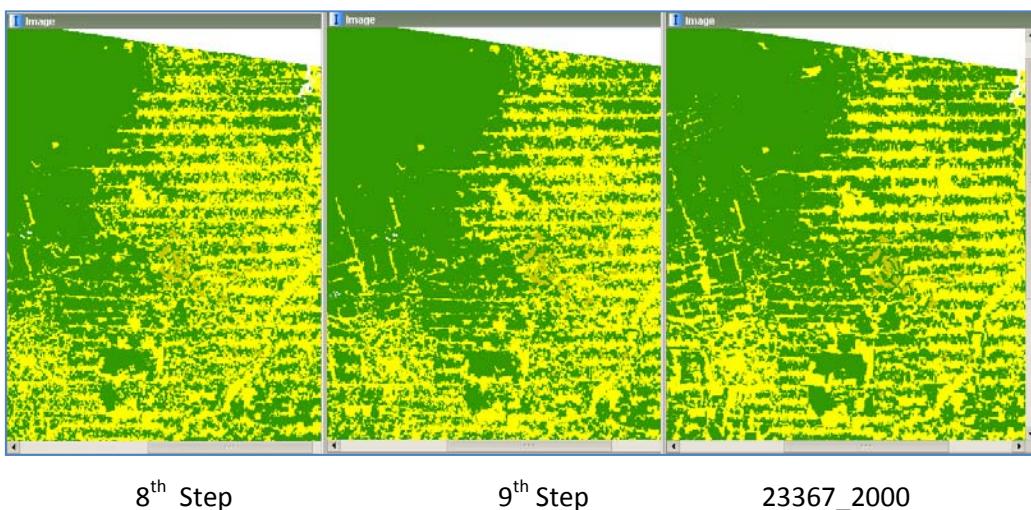


Assim, *Patcher* receberá o mapa de paisagem depois de *Expander* ter modificado o mapa e o mapa de probabilidades corroído (corroído) (ajuste para 0) onde as transições acontecem. No caso de *Expander* não conseguir fazer todas as mudanças especificadas, uma matriz de

mudanças restantes para cada transição será passada para *Add Change Matrix*, que combinará as matrizes recebidas de *Modulate Change Matrix* e do porto **Remaining Changes** do *Expander*. Seus demais parâmetros são ajustados como aqueles do *Patcher*.



Agora execute o modelo, acesse o log para o relatório, e compare os resultados com os resultados anteriores.



Este modelo se aproxima mais da estrutura da paisagem final observada?

Tente mudar os parâmetros de *Modulate Change Matrix*, *Expander* e *Patcher* para ver o que você obtém. **DICA:** mude apenas um parâmetro por vez. Agora que o modelo está calibrado tanto em relação à localização das mudanças quanto à estrutura da paisagem, assim como está validado, você pode aplicá-lo para fins de projeção futura. Vamos avançar para o último passo.

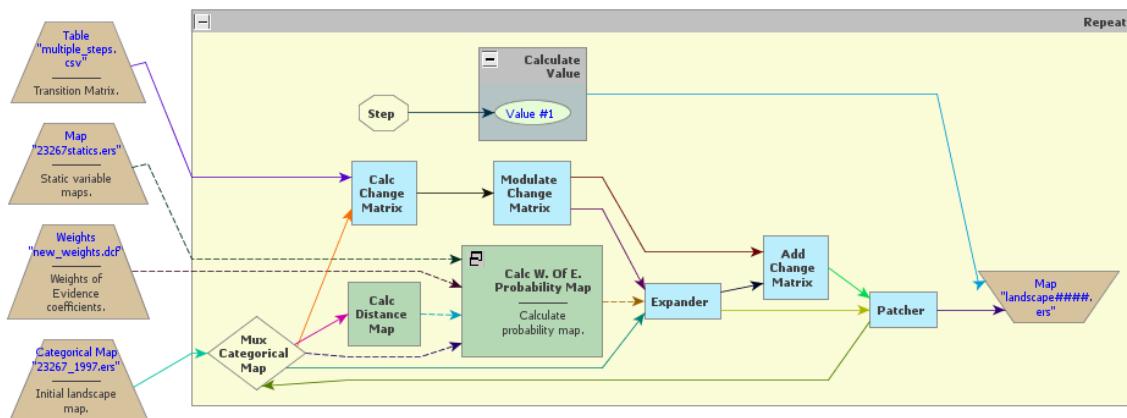
7.10 Décimo passo: Projetando trajetórias de desmatamento

Modelos de simulação podem ser entendidos como um dispositivo heurístico útil para avaliar, em curto e longo prazo, os resultados de uma variedade de cenários, traduzidos como diferentes conjunturas socioeconômicas, políticas e ambientais. Entre eles, uma classe especial de modelos, os modelos espacialmente explícitos, simula a dinâmica de um sistema ambiental, reproduzindo a maneira como seus padrões espaciais evoluem, para projetar as consequências ecológicas e socioeconômicas prováveis decorrentes da dinâmica do sistema.

Assim, como é mostrado através do presente exemplo, podemos aplicar o modelo de simulação para estimar os impactos de trajetórias futuras de desmatamento sob vários cenários socioeconômicos e de políticas públicas, relativos às emissões de gases do efeito estufa (Soares-Filho *et al.*, 2006), mudanças climáticas regionais (Schneider *et al.*, 2006;

Sampaio *et al*, 2007), regime hídrico (Costa *et al*, 2003, Coe *et al*, 2009), perda de habitat e fragmentação de ecossistemas (Soares-Filho *et al.*, 2006, Texeira *et al.*, 2009), assim como a perda de serviços ambientais florestais e bens econômicos (Fearnside, 1997).

Abra o modelo simulate_deforestation_from_1997_2000_30years_ahead.xml disponível em \Examples\setup_run_and_validate_a_lucc_model\10_run_deforestation_trajectories e mantenha os functores Group abertos.



As únicas diferenças entre este modelo e o modelo anterior consistem na sua paisagem inicial, que é agora um mapa do ano 2000, o número de iterações ajustado para “30”, e três functores adicionais *Calculate Value*, *Number Value*, e *Step* empregados para fornecer o sufixo para o nome da paisagem simulada nos anos seguintes ao ano 2001. Além disso, para economizar espaço, o mapa de probabilidade não será salvo.

DICA: você precisa mover o mapa “Landscape####.ers” para fora do *Repeat* para quebrar a ligação do **Step**.

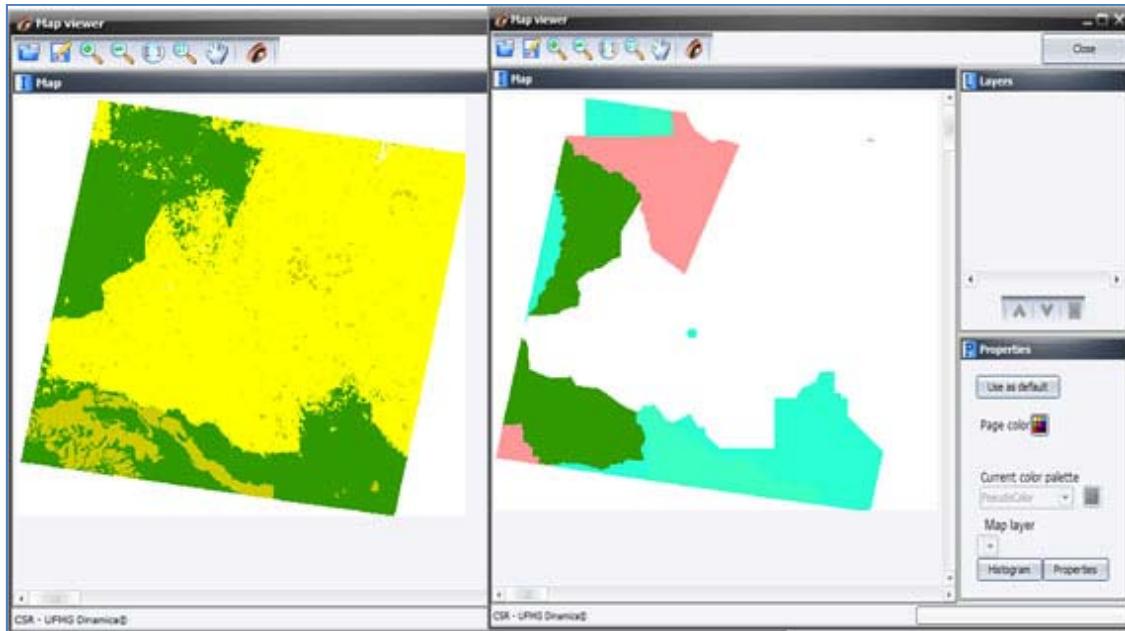


Como este modelo usa taxas fixas de transição, nós podemos considerar que ele projeta a tendência histórica para o futuro, por isso, ele foi nomeado como cenário de tendência histórica.

Verifique as diferenças entre este modelo e o anterior abrindo os functores e suas conexões e, em seguida, execute o modelo e abra o mapa “landscape2030.ers”. **DICA:** Talvez você queira salvar apenas o mapa de paisagem final. Para fazer isso, você precisa apenas arrastar *Save Map* para fora do *Repeat*.



Isto é o que você obteve:



Observe que a floresta desaparece virtualmente fora das áreas protegidas, as quais começam a serem invadidas. Isto é o que irá acontecer se a tendência histórica recente de desmatamento continuar pelo futuro.

Mapas animados são poderosas ferramentas para conscientizar a população em geral sobre as possíveis consequências de alterações na dinâmica do sistema, tal como o desmatamento na Amazônia. **DICA:** Salve os mapas resultantes a cada passo de tempo no formato Geotiff. Em seguida, você pode importá-los para um SIG ou para um software de processamento de imagens para construir animações.

Você pode querer melhorar este modelo incorporando taxas dinâmicas, outras variáveis dinâmicas, submodelos, tais como o construtor de rodovias, e retroalimentações a partir dos atributos da paisagem para o cálculo da taxa de transição. Tudo isso é possível no Dinamica EGO. Um exemplo de módulo de construtor de rodovias é encontrado no modelo `mato_grosso_road.xml`

em
\Examples\run_lucc_northern_mato_grosso\run_roads_with_comments. Além disso, Dinamica EGO permite a incorporação de cenários econômicos, sociais e políticos em um modelo que integra o efeito dessas causas subjacentes (Geist, & Lambin, 2001) na trajetória do desmatamento (normalmente estas variáveis são entradas no modelo na forma de tabelas com chaves especificadas para uma unidade geográfica, tal como um município, estado ou país). Por fim, Dinamica EGO também permite a união com modelos externos desenvolvidos no VENSIM, um software de pensamento sistêmico (www.vensim.com). Por exemplo, VENSIM pode ser usado para modelar o efeito de um cenário complexo de taxas de transição. Parte dessas características avançadas será demonstrada em lições avançadas e em um exemplo da aplicação do Dinamica EGO na modelagem de cenários para um estudo de caso REDD (Redução de Emissões do Desmatamento e Degradação Florestal). Mas antes de entrar nessas lições, vamos aprender algumas características que são essenciais para o desenvolvimento de modelos complexos no Dinamica EGO.

8. Linguagem de script e execução em console do Dinamica EGO

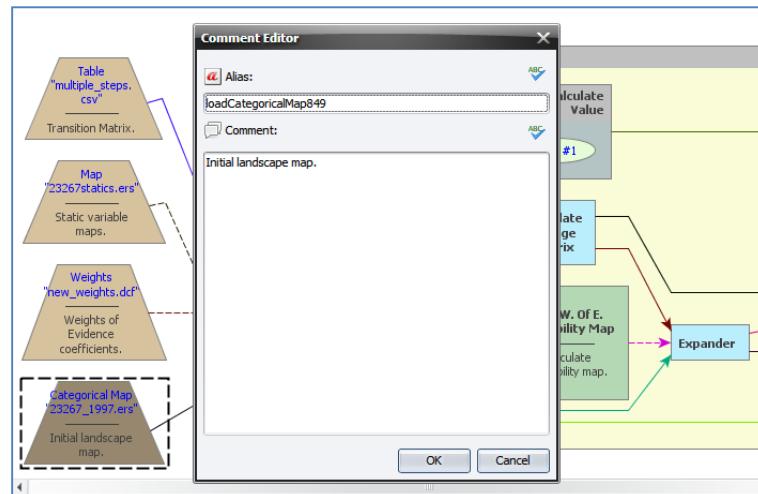
O que você irá aprender?

- Nomeando variáveis com alias (apelidos)
- Linguagem de programação do Dinamica EGO
- Execução em console do Dinamica EGO

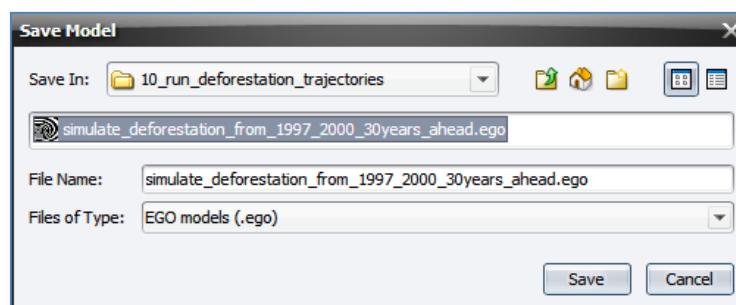
À medida que um modelo se torna mais e mais complexo, você provavelmente achará útil salvá-lo no formato EGO para continuar desenvolvendo o modelo usando a linguagem de programação EGO em um editor de texto (e.g. NotePad++, Context ou compilador C++). Por exemplo, o modelo da Extração de Madeira na Amazônia (Merry *et al.*, 2009), desenvolvido no Dinamica EGO envolve mais de mil linhas de código de programação. Ambos os formatos script (XML, EGO) são 100% compatíveis, de forma que os usuários podem trazer o modelo e retornar com ele da interface gráfica para o editor de texto sem perder nenhuma informação. Os modeladores avançados irão se beneficiar bastante desta linguagem de programação bem estruturada e de fácil uso. **DICA:** O formato EGO é também um bom jeito de unir dois modelos, uma vez que você pode copiar e colar partes de seus scripts.

Abra novamente o modelo “simulate_deforestation_from_1997_2000_30years_ahead.xml”. Para tornar o modelo mais fácil de entender no editor de texto, você precisa primeiro nomear algumas variáveis importantes e depois adicionar comentários às partes do modelo. Vamos fazer isso. Selecione a ferramenta Add Comment to Functor e clique no mapa categórico “23267_1997.ers”.

Escreva no campo alias “initial_landscape”, clique no *Calc Distance Map* e escreva no alias “distance_to_deforested”.



Faça o mesmo para os functores: Map “23267statics.ers”, *Calc W. Of. E. Probability Map*, *Calculate Value*, *Expander*, *Patcher*, e *Mux Categorical Map*, escrevendo os respectivos alias: “statics”, “probability”, “year”, “landscape_with_patches_expanded”, “landscape_with_new_patches”, e “feed_back_landscape”. Agora salve o modelo no formato EGO.



Abra o modelo “simulate_deforestation_from_1997_2000_30years_ahead.ego” em um editor de texto (de preferência um que destaque o script no formato C++, e.g. Notepad++, disponível em <http://notepad-plus.sourceforge.net>). O script EGO é como o seguinte:

```
/**  
dff.date = Fri Sep 04 12:57:13 2009  
dff.version = 1.3.4.20090827-beta  
  
metadata.author = Dinamica Team  
  
metadata.description = This script corresponds to a deforestation model. Only one transition is modeled: 2 to 1, time-period comprises 2000 to 2030, divided in annual time steps. The land cover classes are non-forest (3), forest (2), and deforested (1). The model is set to form patches and to expand previous patches of deforested land.  
  
metadata.organization = CSR / UFMG  
  
metadata.showproperties = yes  
  
metadata.title = Simulate Deforestation  
  
*/  
  
Script {{  
  
    // Initial landscape map.  
  
    initial_landscape := LoadCategoricalMap {  
        filename = "../originals/23267_1997.ers",  
        loadAsSparse = .no,  
        defineNullValue = .no,  
        nullValue = 0,  
        suffixDigits = 0,  
        step = .none  
    };  
  
    // Static variable maps.  
  
    statics := LoadMap {  
        filename = "../originals/23267statics.ers",  
        loadAsSparse = .no,  
        defineNullValue = .no,  
        nullValue = 0,  
        suffixDigits = 0,  
        step = .none  
    };  
  
    // Weights of Evidence coefficients.  
  
    loadWeights851 := LoadWeights "../4_weights_of_evidence_correlation/new_weights.dcf" 0 .none;  
  
    // Transition Matrix.  
  
    loadLookupTable852 := LoadLookupTable "../1_transition_matrix_calculation/multiple_steps.csv" 0 .none;  
  
    // Simulation model.  
  
    @collapsed = no
```

Este é o cabeçalho

Aqui o modelo começa

Aqui o mapa categórico é carregado

functor é envolvido por {{ }}

Alias substitui nomes internos

e.g. initial_landscape

Comentários depois de //

```

Repeat 30 {{
    step = step;

    feed_back_landscape := MuxCategoricalMap initial_landscape landscape_with_new_patches;

    modulatedChanges complementaryChanges := ModulateChangeMatrix (CalcChangeMatrix feed_back_landscape
loadLookupTable852) [
        2->1 0.2
    ];

    distance_to_deforested := CalcDistanceMap {
        categoricalMap = feed_back_landscape,
        categories = [ 1 ],
        cellType = .int32,
        nullValue = -2147483648,
        truncateDistance = .yes
    };

    // Calculate probability map.

    @collapsed = yes
    probability := CalcWOfEProbabilityMap {
        landscape = feed_back_landscape,
        weights = loadWeights851,
        transitions = [ 2->1 ],
        cellType = .uint8,
        nullValue = 0
    }{{
        NameMap statics "static_var";

        NameMap distance_to_deforested "distance";
    }};
}

@alias = landscape_with_patches_expanded
changedLandscape corrodedProbabilities remainingChanges := Expander {
    landscape = feed_back_landscape,
    probabilities = probability,
    changes = modulatedChanges,
    transitionParameters = [
        2->1 20 50 1.5
    ],
    neighborWindowLines = 3,
    neighborWindowColumns = 3,
}

```

Container *Repeat* começa

Expander recebe `feed_back_landscape`

```

pruneFactor = 10
};

step861 := Step step;

landscape_with_new_patches __:= Patcher {
    landscape = changedLandscape,
    probabilities = corrodedProbabilities,
    changes = (AddChangeMatrix remainingChanges complementaryChanges),
    transitionParameters = [
        2->1 25 50 1.5
    ],
    neighborWindowLines = 3,
    neighborWindowColumns = 3,
    pruneFactor = 10
};

year := CalculateValue [
    2000 + v1
] .no 0 ^{
    NumberValue step861 1;
};

SaveMap {
    map = landscape_with_new_patches,
    filename = "landscape.ers",
    suffixDigits = 4,
    step = year,
    useCompression = .yes
};
};

}};


```

Patcher recebe `changed_landscape`

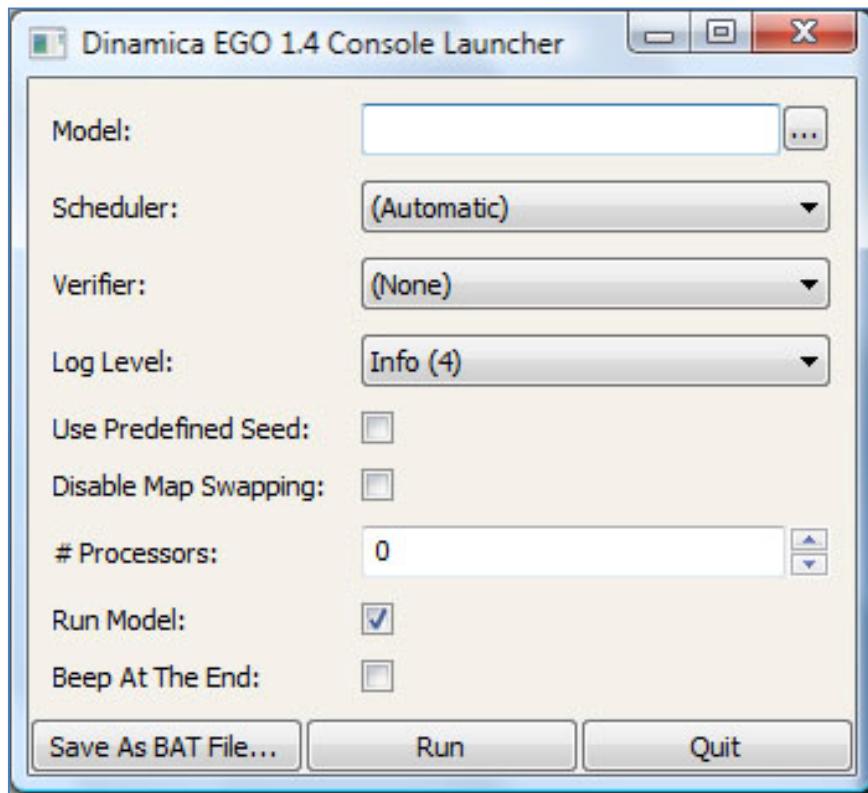
Functor `SaveMap`

Fecha-se Container `Repeat`

Fecha-se o modelo

Agora que você aprendeu um pouco sobre como programar na linguagem de script EGO, outro recurso disponível no Dinamica EGO é o Console Launcher (Execução em console). Qualquer modelo salvo no Dinamica EGO, seja ele XML ou EGO, pode ser executado a partir do prompt de comando. Executar o modelo a partir do prompt de comando aumenta a performance do modelo, pois o modelo se torna livre do peso representado pela interface gráfica. Além disso, a partir do prompt de comando Dinamica EGO tira vantagem da arquitetura de multiprocessadores, ou dividindo os caminhos de execução em diferentes processadores ou rodando muitos dos algoritmos internos dos functores usando processamento paralelo. Dessa forma, a execução se torna bem mais rápida. Dependendo da quantidade de dados carregados na memória, é também possível desabilitar a troca de dados em disco. **DICA:** O Windows 32 bits consegue lidar somente com até 3 gigabytes de memória para um único processo.

Para facilitar o processo de execução de um modelo a partir do comando do prompt, nós desenvolvemos uma ferramenta de Console Launcher. Seu uso é fortemente recomendado para modelos complexos e outros com grande quantidade de dados. Chame o Console Launcher da subpasta de ferramentas no atalho Start/Programs/Dinamica EGO. Ele irá abrir a seguinte interface:



Model: nome do modelo, **Scheduler:** como os functores do modelo serão sequenciados, **Verifier:** tipo de verificação de script, **Log Level:** nível máximo de relatório de Log, **Use Predefined seed:** marque esta caixa para usar uma semente predefinida para gerar números aleatórios, **Processors:** número de processadores que o modelo irá usar. Use 0 para detectar automaticamente o número disponível. **Run Model:** Marque esta caixa para executar o modelo completamente. Você também pode salvar a configuração do Console Launcher em um arquivo batch. Isso é útil caso você quiser acoplar Dinamica EGO com outro software. **DICA:** outro jeito de acoplar livremente outros programas de computador com Dinamica EGO é através do functor *Run External Process*, disponível na aba Control. Usando este functor, você será capaz de chamar um programa externo a partir do Dinamica, passar a este programa resultados intermediários e alimentar, novamente, o modelo do Dinamica EGO com os resultados obtidos.

Parabéns, você já foi longe! Mas ainda há numerosas possibilidades a explorar no desenvolvimento de modelos com Dinamica EGO. Para aqueles que procuram por habilidades mais avançadas em modelagem nós apresentamos o próximo capítulo, que contem exemplos usando recursos avançados.

9. Recursos Avançados

O que você irá aprender?

- Variando parâmetros em uma simulação
- Usando sub-regiões em um modelo de simulação
- Usando o conceito de tempo de permanência
- Usando transições determinísticas
- Usando saturação local
- Functores:
 - *Select Transition Matrix*
 - *Calc Change Matrix*
 - *Modulate Change Matrix*
 - *Add Change Matrix*
 - *Select Weights*
 - *Extract Map Layer*
 - *Create Cube Map*

9.1 Variando parâmetros em uma simulação

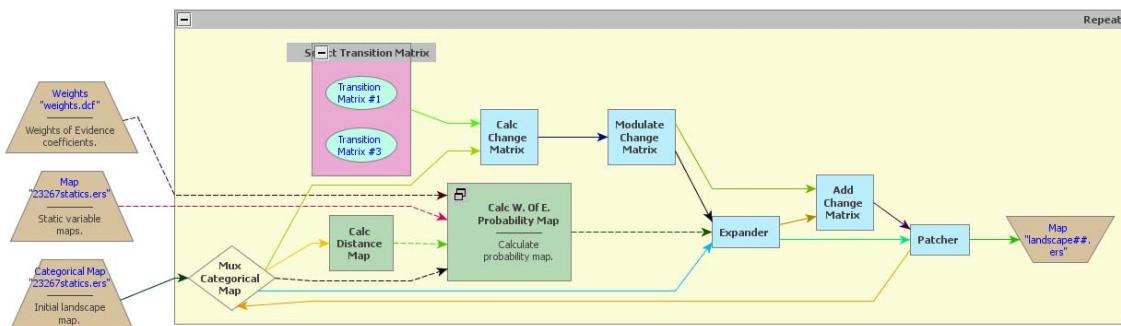
Nada permanece constante em um modelo dinâmico. Considerando isso, Dinamica EGO permite variar os parâmetros do modelo também em diferentes fases ou a cada passo de tempo. Na aba Stack, um conjunto de functores Select pode ser usado para mudar os valores de parâmetros de entrada de acordo com o passo do modelo.

Usando diferentes matrizes de transição

Por exemplo, em vez de usar apenas uma matriz de transição constante, *Select Transition Matrix* permite o uso de diferentes matrizes de transição em uma simulação.

Para começar, carregue o modelo

"simulate_deforestation_using_multiple_transition_matrix.xml" da pasta \\ advanced\\multiple_parameters\\simulate_deforestation_using_multiple_transition_matrix



O uso de mais de uma matriz de transição em um modelo é bem simples. Nesta versão de um modelo de desmatamento, *Select Transition Matrix* foi adicionado para permitir a escolha de uma matriz de transição de acordo com o passo do modelo. Cada matriz de transição é inserida em um *Number Transition Matrix*, clicando nesse functor com o Edit Functor.



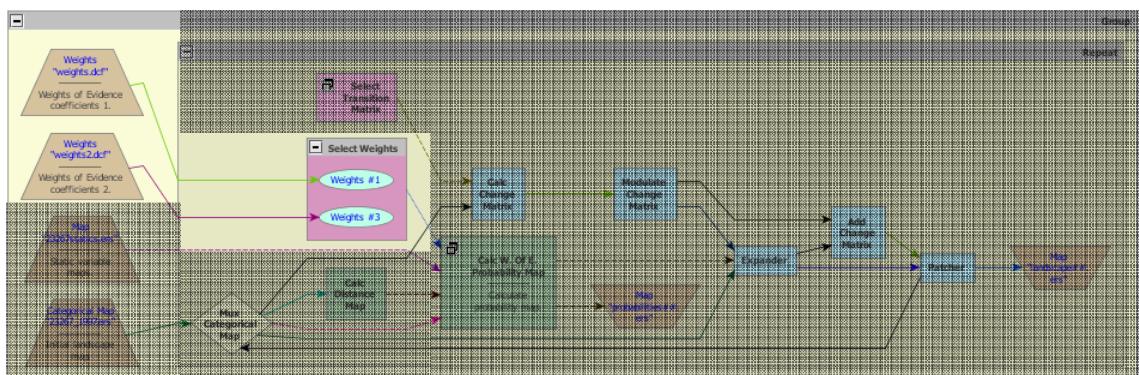
Matrix Number, nesse functor, identifica o passo do modelo no qual essa matriz de transição começará a operar. O modelo usará esta matriz até encontrar um *Number Transition Matrix* com **Matrix Number** que corresponde ao passo atual; deste ponto do modelo em diante, o *Select Transition Matrix* muda para a nova matriz de transição. **DICA:** você pode usar tantas matrizes de transição quanto o número de passos de tempo. Você precisa apenas adicionar mais functores *Number Transition Matrix* e numerá-los de acordo com o intervalo de tempo do modelo.

Usando diferentes coeficientes de Pesos de Evidência.

O mesmo procedimento se aplica à seleção de outros parâmetros do modelo, tais como os coeficientes dos Pesos de Evidência.

Carregue o modelo: "simulate_deforestation_using_multiple_transition_matrix_and_weights.xml" da pasta \Examples\advanced\multiple_parameters\simulate_deforestation_using_multiple_transition_matrix_and_weights

Um functor *Select Weights* é adicionado ao modelo para permitir a seleção de mais de um arquivo de coeficientes de Pesos de Evidência. Um *Number Weight* é inserido dentro do container *Select Weights* para cada arquivo de coeficientes. Outro *Load Weights* é adicionado ao modelo na parte de fora do loop principal (*Repeat*). Este functor carrega os coeficientes de um arquivo e é conectado a um dos functores *Number Weights* que estão dentro do *Select Weights*. Abra cada um dos functores *Number Weights* e edite o campo **Weight Number** para definir o passo no qual cada arquivo de coeficiente começará a tomar parte no modelo.



9.2 Usando sub-regiões em um modelo de simulação

Carregue o modelo: “simulate_deforestation_using_sub_regions.xml” da pasta Examples\advanced\sub_regions\simulate_deforestation_using_sub_regions

O conjunto de functores Subregion é usado para dividir um mapa em partes para processar cada conjunto de dados das sub-regiões separadamente e, em seguida, combinar os resultados novamente (fig. 15). Fazendo uso da abordagem de sub-regiões, você pode definir uma sequência de operações que serão aplicadas apenas a certas sub-regiões ou estabelecer diferentes parâmetros e coeficientes para cada sub-região, modelando como resultado o contexto regional que influencia um fenômeno particular.

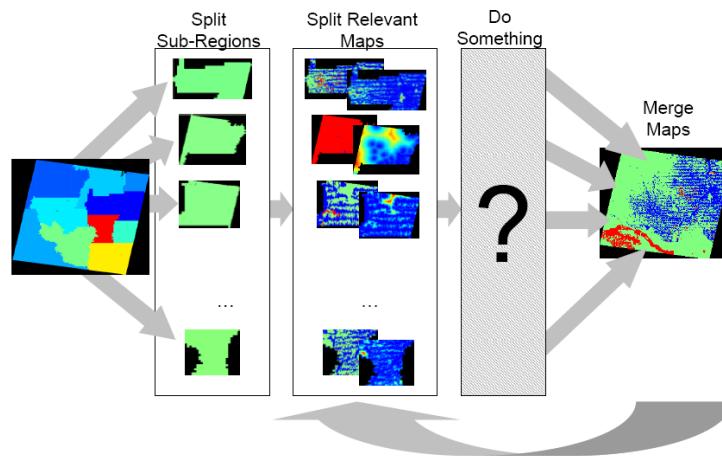
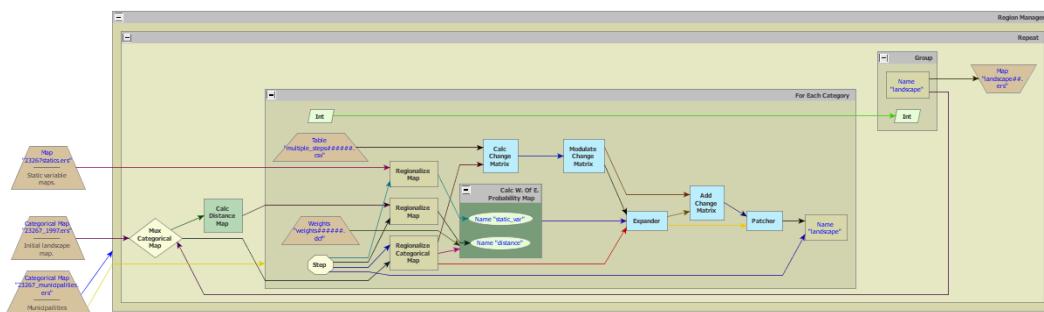


Fig. 15 – O esquema de sub-regiões permite a aplicação de diferentes submodelos para partes de um mapa.

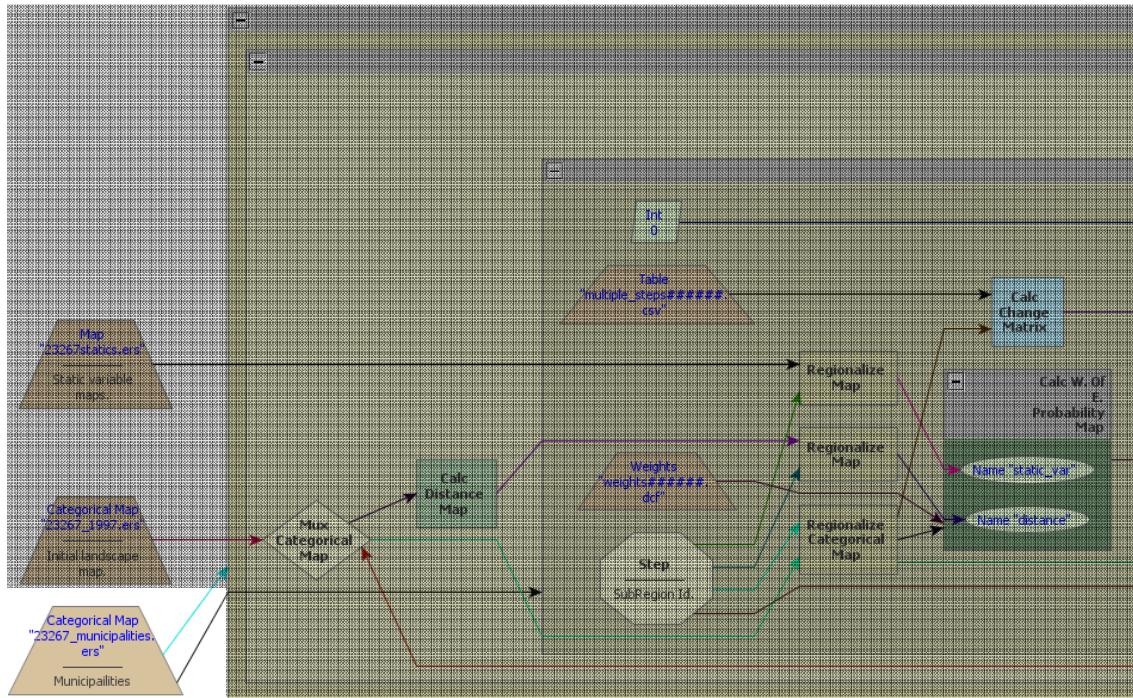
Neste modelo, nós introduzimos sub-regiões em uma simulação para permitir o uso de diferentes matrizes de transição e coeficientes de Pesos de Evidência para cada sub-região de um mapa.



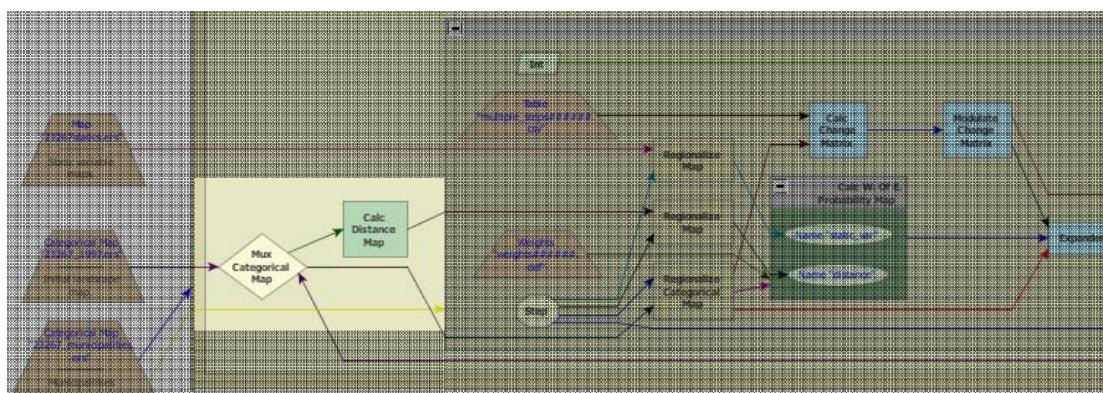
Para fazer isso, você precisa modificar ligeiramente o modelo de simulação da lição 7. Primeiro, uma calibração específica para cada sub-região deverá ser feita. No exemplo abaixo, as matrizes de transição e um conjunto de coeficientes de pesos de evidência foram calculados para cada uma das sub-regiões do mapa. Veja modelos para calibração de sub-regiões diferentes em “Examples\advanced\sub_regions\calibration”.

No modelo de simulação, um *Load Categorical Map* deverá ser adicionado. Ele será responsável por carregar o mapa com as sub-regiões. **DICA:** o mapa deve ser do tipo categórico (neste exemplo, um mapa de limites municipais). A área de cada município é

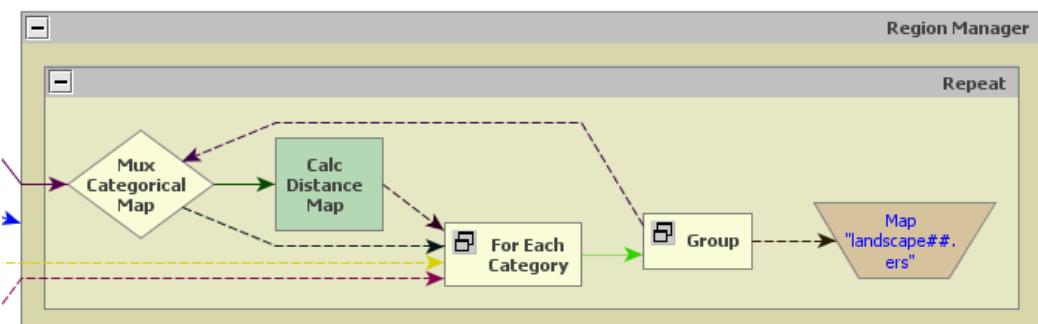
identificada por um código de 6 dígitos. Este código irá identificar as sub-regiões e controlar o processo de regionalização.



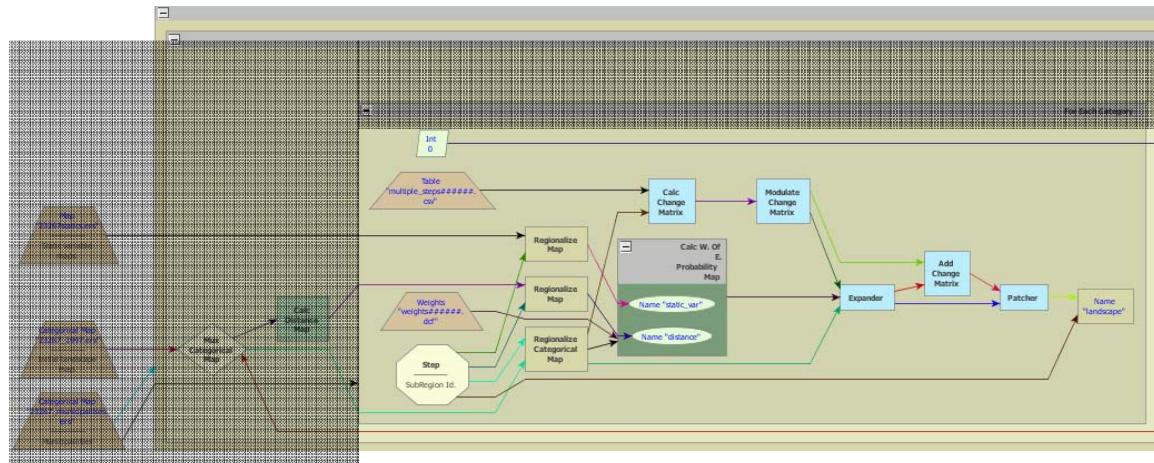
Neste modelo, o desmatamento é simulado em base anual para cada um dos municípios. Depois disso, os mapas resultantes são combinados em um novo mapa de uso da terra que será dividido novamente no começo do próximo passo. Para assegurar continuidade espacial através das sub-regiões, algumas operações, tais como o cálculo do mapa de distância até células desmatadas usando o *Calc Distance Map*, são realizados para o mapa inteiro.



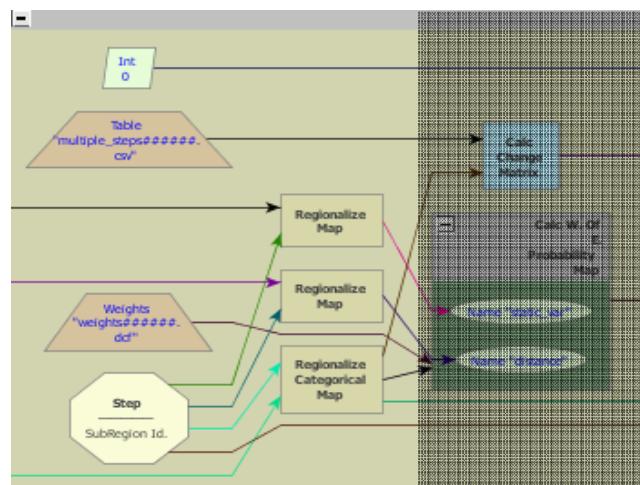
Um *Region Manager* é adicionado fora do *loop* principal (*Repeat*). Este container controla a criação e a união de sub-regiões.



O container *For Each Category* também controla o processo de criação de sub-regiões. O functor *For Each Category* irá repetir a sequência de functores dentro dele. Certifique-se de que o mapa categórico “23267_municipalities.ers” está conectado ao *For Each Category*, assim como ao *Region Manager*. Neste caso, *Step* colocado dentro do container mais interno recebe e passa os códigos das sub-regiões. Além disso, qualquer sequência de functores que afetam cada região deverá ser colocada dentro desse container.

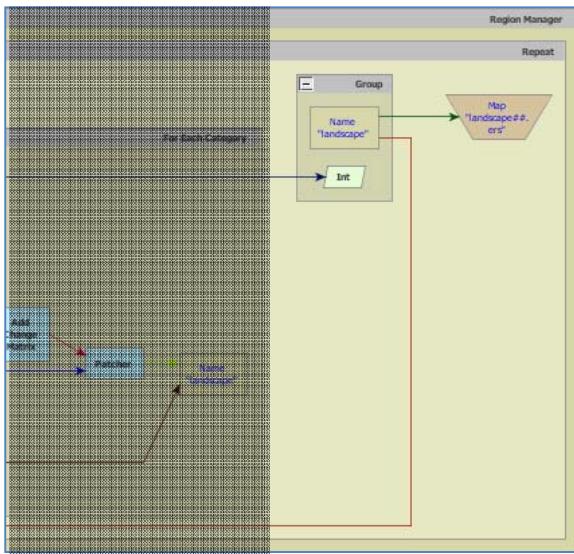


Os functores *Regionalize Map* e *Regionalize Categorical Map* dividem o mapa em mapas de região. Certifique-se de que o *Step* está conectado a eles. Todos os mapas que são combinados depois destes functores devem ter as mesmas dimensões. Portanto, certifique-se de que todos eles estão regionalizados através desses functores, como o exemplo dos mapas de distância e as variáveis estáticas. Observe também que você pode inserir matrizes de transição ou arquivos de Pesos de Evidência para as sub-regiões apenas utilizando o código das sub-regiões como sufixo para esses arquivos e inserindo um *Load Table* e um *Load Weights* dentro do container *For Each Category*. **DICA:** você também precisa mudar o número de dígitos do sufixo do arquivo para 6.



Ao final de uma iteração do *For Each Category*, um functor *Regional Categorical Map* armazena os mapas regionais. Dê um **Global Map Name** para estes mapas. Depois que o *For Each Category* rodar, esses mapas regionais podem ser unidos em um mapa de paisagem atualizado. O functor *Merge Regional Categorical Map* é encarregado de mosaicar os mapas de região. Seu **Global Map Name** deve ser o mesmo do *Regional Categorical Map*. O functor *Merge Regional Categorical Map* deve ser colocado dentro do *Region Manager* e, para assegurar uma sequência adequada de processamento, você precisa estabelecer uma relação de dependência entre esse functor e *Regional Categorical Map*. Um modo de fazer isso é

colocando um functor *Merge Regional Categorical Map* dentro de um *Group* e ligar este *Group* a *For Each Category* usando um functor *Int* como ligação. O *Int* simplesmente passa uma constante inteira do *For Each Category* para o *Group*, estabelecendo, como resultado, uma dependência temporal entre as execuções de ambos.



Depois dos mapas de região serem unidos, o modelo itera e um novo mapa de paisagem retorna para alimentar o *Mux Categorical Map* fechando o loop (ciclo completo). Portanto, o mapa de paisagem salvo representa um mosaico de paisagens regionais alteradas.

9.3 Usando o tempo permanência

Carregue o modelo “simulate_deforestation_and_abandonment_using_sojourn_time.xml” da pasta \Examples. O uso do tempo de permanência implica que uma transição específica ou evento não ocorrerá a menos que tenha se passado certo tempo desde a ocorrência de um evento específico.

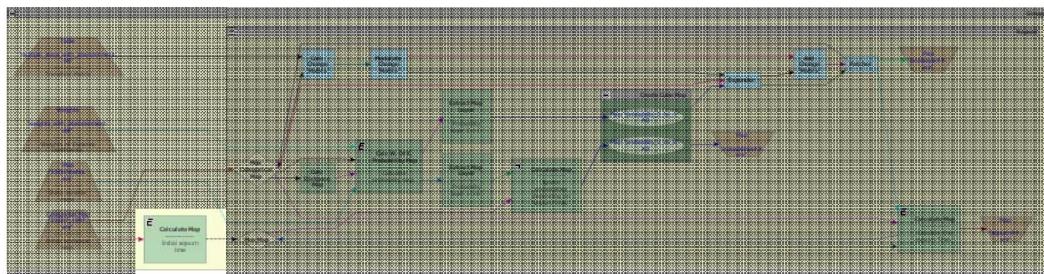
O conceito de tempo permanência pode ser implementado usando um mapa que mantém o registro do tempo decorrido desde a ocorrência de um evento particular (por exemplo, uma transição do estado *i* para *j*). Para cada célula, um relógio irá contar o tempo que passou depois da ocorrência do evento. No final da iteração, este mapa é atualizado: se não há nenhum evento novo, o modelo simplesmente adiciona 1 à célula do tempo de permanência, e se um evento ocorre, o relógio é ajustado para zero (fig. 16).

Landscape			Sojourn Time		
t0	t1	t2	t3		
1 2 2	1 1 2	1 1 1	1 1 1		
3 1 2	3 1 2	3 1 2	3 1 1		
X 2 1	X 1 1	X 1 1	X 1 1		

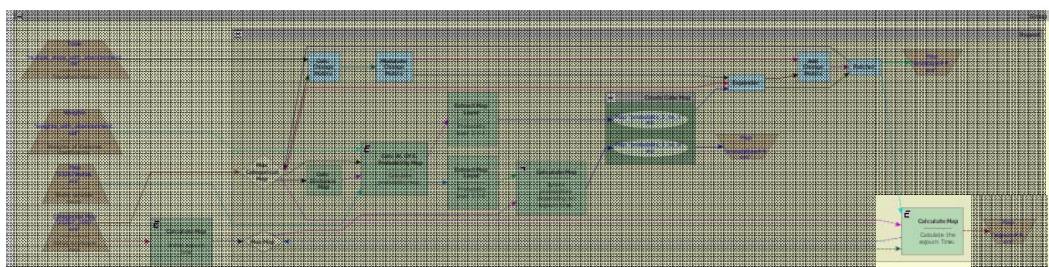
Fig. 16 – Neste caso, o mapa do tempo permanência mantém o registro do tempo decorrido depois de uma célula ter sido alterada para um novo estado. Todas as células começam com 99 unidades de tempo.

O tempo de permanência pode ser usado como uma regra para restringir uma transição de um estado para outro apenas depois de ter havido certo tempo de permanência em um estado particular. Neste exemplo, o tempo de permanência é usado para influenciar as probabilidades da mudança de terra desmatada (valor 1) para abandono – revegetação (valor 8). A terra só é abandonada depois de 4 anos em uso, isto é, depois que o desmatamento se estabeleceu (transição 2 para 1).

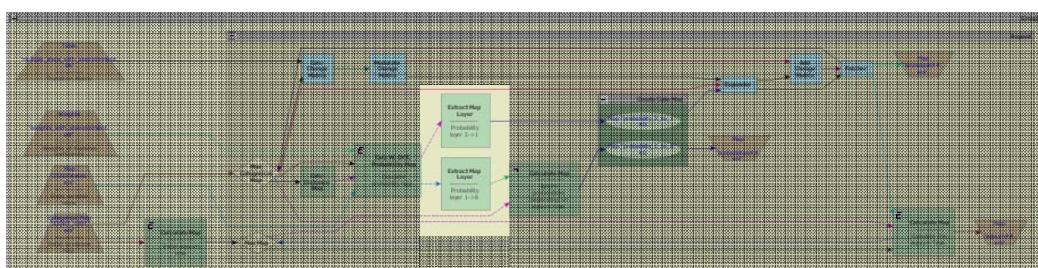
O mapa de tempo de permanência é inicializado com altos valores (e.g. 9999). Fazendo isso, é possível diferenciar células onde certa transição nunca ocorreu e células que mudaram durante a simulação. Isto pode vir a ser útil para análises adicionais.



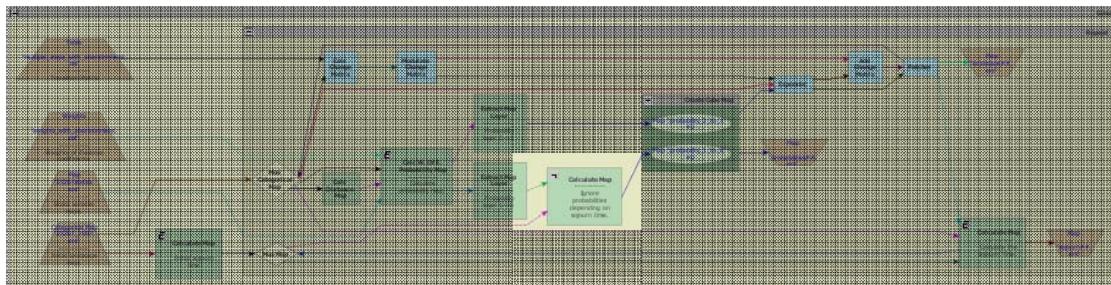
Ao final da iteração, um novo mapa de tempo de permanência é calculado comparando o mapa anterior de uso da terra com o atual. As células cujos estados foram alterados, recebem zero em seus tempos de permanência correspondentes; do contrário, seus valores são incrementados em 1 a cada unidade de tempo.



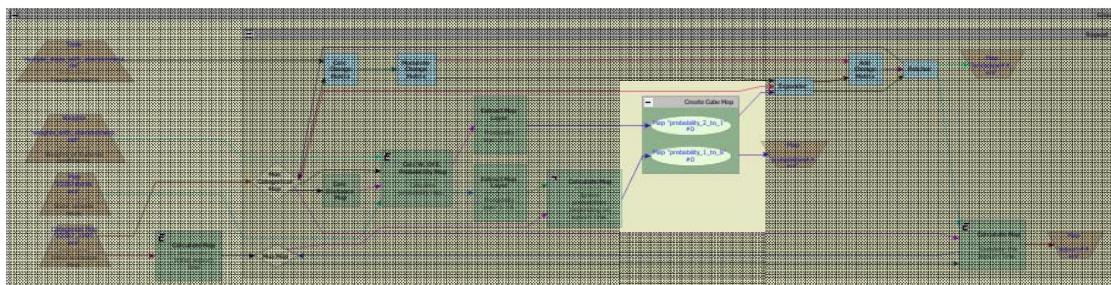
O conceito do tempo de permanência é também aplicado para restringir uma mudança. Um jeito fácil de fazer isso é modificando o mapa de probabilidade correspondente. Como esses mapas são empilhados em um cubo raster, você precisa primeiro extrair as camadas do conjunto de raster, e então, o *Calculate Map* conseguirá manipulá-los separadamente.



Depois de calcular o mapa de probabilidade usando o *Calc W. of E. Probability Map*, a camada da transição de desmatamento para abandono (transição 1 para 8) passa por um novo cálculo, uma comparação do tempo de permanência. Se uma célula desmatada tem mais de 3 anos, sua probabilidade é mantida inalterada. Se não, sua probabilidade de alterar para o estado de abandono é ajustada para zero, impedindo, assim, a ocorrência dessa transição. **DICA:** Os functores de transição (*Patcher* e *Expander*) não alteram células com probabilidades de transição iguais a zero.



Em seguida, as camadas das probabilidades (a camada original com transição de 2 para 1 e a nova camada com a transição de 1 para 8) são unidas novamente em um novo conjunto de cubo raster que é passado aos functores de transição, *Expander* e *Patcher*.



9.4 Usando o tempo de permanência e transições determinísticas

Carregue o modelo “simulate_def_aband_and_deterministic_transition.xml” da pasta
\\Examples\\ sim_defor_and_aband_using_sojourn_time_and_deterministic_transition

Neste exemplo, a transição de abandono (a terra abandonada sofre regeneração) para floresta é feita deterministicamente baseada apenas no tempo de permanência. Assim, após um período em regeneração, a parcela abandonada adquire uma estrutura florestal similar à floresta madura, sendo assim, considerada como floresta. O tempo de permanência mínimo para essa transição ocorrer é de 21 anos (fig. 17).

1	2	2	1	2	2	1	2	2	1	2	2
3	8	1	3	8	1	3	8	1	3	2	1
X	2	8	X	2	8	X	2	2	X	2	2
Landscape											
119	119	119	120	120	120	121	121	121	122	122	122
119	18	5	120	19	6	121	20	7	122	0	8
X	119	19	X	129	20	X	1	0	X	2	1

Sojourn Time

t20 t21 t22 t23

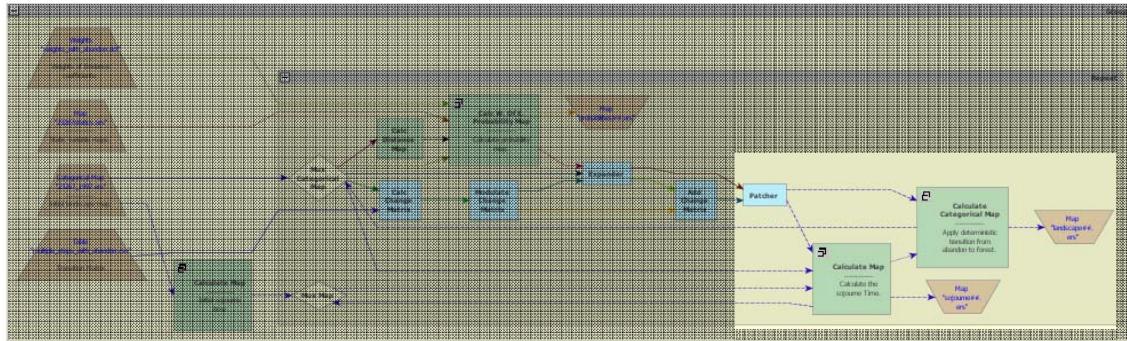
Transition 8 to 2 occurs only if $t(\text{cell}) > 20$

Fig. 17 – A transição de terra abandonada para floresta é feita deterministicamente apenas baseada no tempo de permanência.

Uma transição adicional de terra desmatada para abandono é adicionada ao modelo anterior. Um *Calculate Categorical Map* colocado ao final do modelo é empregado para fazer essa transição. A regra é a seguinte:

if i1 = 8 and i2 > 20 then 2 else i1.

$i1$ é o mapa de paisagem e $i2$ o mapa de permanência (floresta é igual a 2, terra abandonada é igual a 8). Observe que, por questão de simplificação, não é possível desmatar áreas em abandono (transição 8 para 1). Isso poderia ser adicionado ao modelo como uma nova transição. Texeira *et al.* (2009) fornece um exemplo de modelagem de dinâmica de paisagem da floresta Atlântica que inclui multi-estados e transições de floresta para vários tipos de usos da terra.



9.5 Usando saturação local

Carregue o modelo “simulate_deforestation_using_local_saturation.xml” da pasta `\Examples\advanced\local_saturation`

A saturação local evita que uma mudança ocorra dentro de uma região específica, na qual a área de uma classe é maior que um limiar estabelecido (fig. 18). Esta característica é útil para simular processo de difusão assim como para se estabelecer uma área mínima de remanescente de floresta (como estabelecido pelo Código Florestal Brasileiro para propriedades privadas). A saturação local pode ser implementada reduzindo a probabilidade de uma transição usando uma função assintótica como mostrado a seguir:

$$P2 = P1 * (Li - Oc) / (Li + Oc), \text{ case } Li \geq Oc$$

$$P2 = 0, \text{ case } Li < Oc$$

onde $P2$ é a nova probabilidade, $P1$ é a probabilidade original, Li é o número máximo de células de uma classe particular onde é permitido ocorrer dentro de certa vizinhança no mapa, e Oc é o número de células desta classe dentro daquela vizinhança.

Neste exemplo, o desmatamento deve parar em uma região local (neste caso, uma janela de 59.25 hectares ou 9 células – uma janela 3x3) quando 50% desta região é desmatada.

Para cada passo do modelo, a quantidade de células desmatadas é calculada em cada janela 3x3 do mapa de paisagem. Neste caso, o limiar consiste de 5 células como se segue:

$$P2 = P1 * (5 - Oc) / (5 + Oc), \text{ case } Li \geq Oc$$

$$P2 = 0, \text{ case } 5 < Oc$$

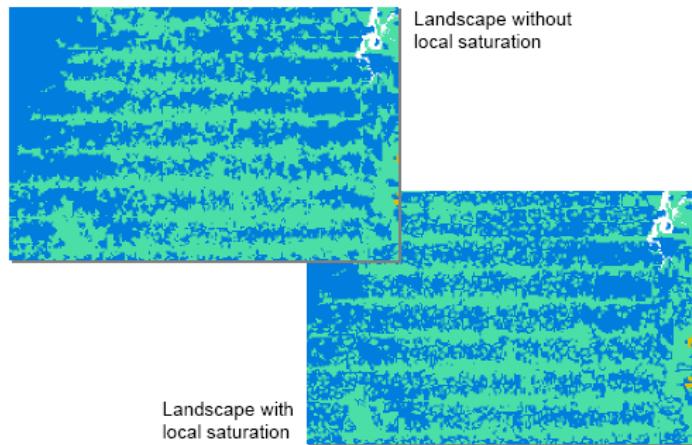


Fig. 18 – Duas paisagens sem e com efeito de saturação

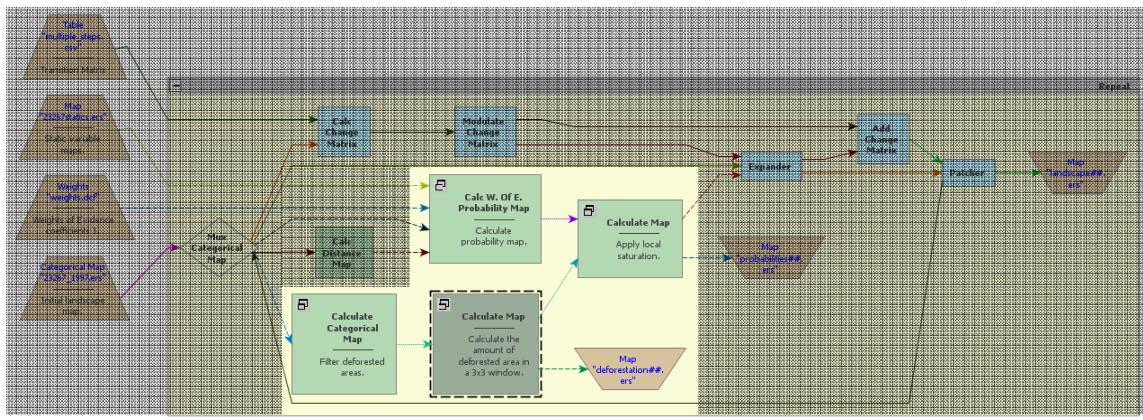
Três functores *Calculate Map* são adicionados no modelo de simulação de desmatamento para incorporar o efeito de saturação. Abra os functores para ver suas equações. O primeiro estabelece “1” para células desmatadas e “0” para floresta. O segundo conta a quantidade de células desmatadas dentro de uma janela de tamanho 3x3. Um operador de vizinhança é utilizado, como se pode ver a seguir:

nbCount(i1, 3, 3)

onde *nbCount* é o operador de contagem de vizinhança, *i1* é o *map # 1*, e 3,3 é o tamanho da janela em células. Você pode facilmente aumentar o tamanho da vizinhança, bastando para isso mudar esses valores. Em seguida, um terceiro *Calculate Map* é adicionado para aplicar a regra de saturação local no mapa de probabilidade como a seguir:

if v1 - i1 >= 0 then i2 * (v1 - i1) / (v1 + i1) else 0

Onde *v1* é o valor de saturação em número de células (neste caso, igual a 5).



Como resultado, o desmatamento irá parar em regiões à medida que alcançam um limiar de saturação e se movem para novas áreas com probabilidades de desmatamento mais baixas, simulando, assim, a difusão do desmatamento.

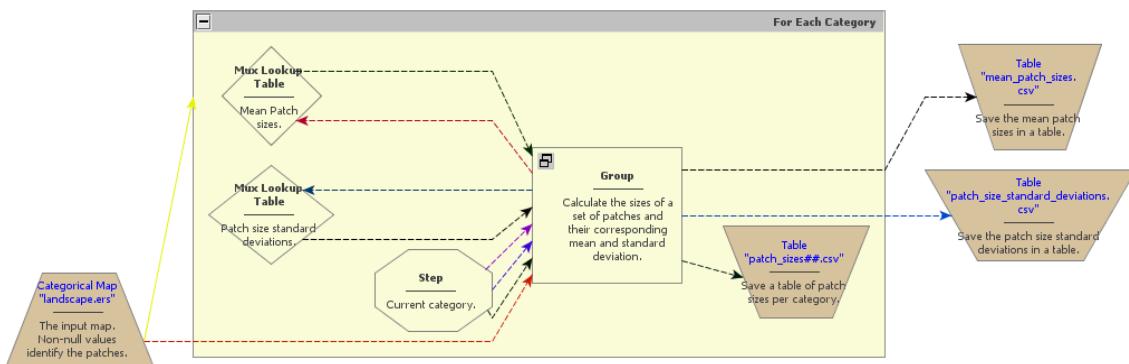
10. Métricas de Paisagem no Dinamica EGO

O que você irá aprender?

- Calcular métricas de paisagem
- Tamanho médio de mancha
- Comprimento médio da borda da mancha
- Dimensão fractal
- Functor:
 - *Log policy*

As Métricas de Paisagem (McGarigal and Marks, 1995) podem ser ferramentas úteis para avaliar a qualidade dos habitats quando extensivos inventários de biodiversidade ou dados ecológicos não estão disponíveis ou são difíceis de obter, uma vez que as métricas de paisagem são fortemente relacionadas aos indicadores de biodiversidade (Metzger, 2006). Por exemplo, métricas de paisagem podem ser aplicadas para identificar a melhor configuração da paisagem para conservação de espécies em florestas — independentemente da percepção por espécies individuais, — que são, hipoteticamente, uma paisagem com: i) uma alta cobertura florestal; ii) um pequeno número de fragmentos florestais; iii) um alto índice da maior mancha florestal que pode suportar populações estáveis e ser uma fonte para pequenas manchas; iv) uma grande área média de mancha florestal; e v) um alto índice médio de proximidade florestal (Texeira et al., 2009). Assim, a aplicação dessas métricas consiste em uma poderosa ferramenta para descrever as consequências das dinâmicas de uso e cobertura da terra na conservação da biodiversidade. Neste contexto, nós introduzimos aqui uma série de modelos desenvolvidos no Dinamica EGO para calcular métricas de paisagem. Ao invés de fornecer uma solução tipo caixa preta todas essas métricas são desenvolvidas usando a linguagem de modelagem do Dinamica EGO e, portanto, elas servem como moldes para se gerar uma grande variedade de outras métricas.

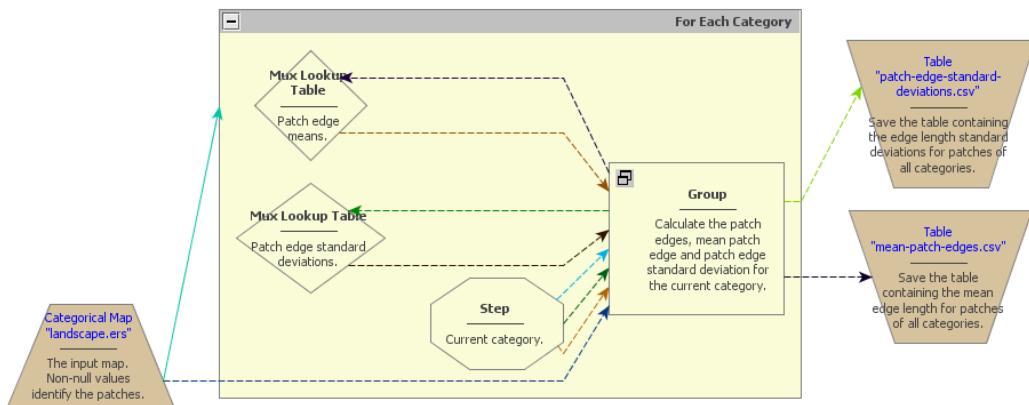
Carregue o modelo “calc_mean_patch_sizes_and_standard_deviations.ego” da pasta Examples\landscape_metrics\calc_mean_patch_sizes_and_standard_deviations



Este modelo calcula o tamanho de cada mancha de uma classe de paisagem e, para cada classe, o tamanho médio de mancha e o desvio padrão do tamanho das manchas. Uma tabela é gerada para cada métrica. Examine este modelo abrindo o functor *Group* e executando o modelo. O functor *Log Policy* presente dentro do *Group* é usado para executar o modelo em modo silencioso, aumentando, assim, a velocidade de processamento.

Carregue agora o modelo “calc_mean_patch_edges_and_standard_deviations.ego” da pasta Examples\landscape_metrics\calc_mean_patch_edges_and_standard_deviations.

Este modelo calcula o comprimento da borda de cada mancha de uma classe de paisagem e para cada classe o comprimento médio da borda de suas manchas e o seu desvio padrão. Examine este modelo abrindo o functor *Group* e, em seguida, execute-o.



Estes dois conjuntos de medidas são usados para o cálculo de métricas mais complexas, tais como dimensão fractal e índice da mancha maior. Por exemplo, a dimensão fractal para uma classe de paisagem pode ser estimada usando a relação perímetro-área, de forma que, se dados suficientes estiverem disponíveis, a declividade da linha obtida executando a regressão de $\log(P)$, os comprimentos das bordas das manchas, em $\log(A)$, as áreas das manchas, é igual a $2/D$ (Burrough, 1986). Verifique outros modelos de métricas de paisagem, tais como distância média de mancha, índice da mancha maior, na pasta Examples\landscape_metrics.

11. Estudo de caso REDD

O que você irá aprender?

- Como projetar taxas de desmatamento baseadas em variáveis socioeconômicas
- Converter taxas brutas em taxas líquidas
- Como desenvolver um modelo de contabilidade de carbono
- Functores:
 - *Calc Neighborhood*
 - *Calc Spatial Lag*

O desmatamento tropical é a segunda fonte antropogênica de GEE (Gases do Efeito Estufa). Sete a vinte por cento das emissões mundiais de CO₂ vem do desmatamento tropical, o que é equivalente a 0,5 a 2,4 bilhões de toneladas de carbono emitidas por ano (Houghton *et al.*, 2005). Uma proposta para compensar países que reduziam suas emissões do desmatamento e da degradação de florestas (REDD) foi inicialmente apresentada pelo IPAM e outras instituições na nona COP do Clima (Conferência das Partes) em Milão, 2003 (Santilli *et al.* 2005; Moutinho & Schwartzman 2005). Esta proposta recomendava que países em desenvolvimento capazes de reduzir o desmatamento para taxas abaixo de uma linha de base histórica sobre um período de tempo estariam aptos a receber compensação financeira da comunidade internacional através de um mercado de créditos de carbono. Mais tarde, esta proposta foi oficialmente apoiada pelos governos dos países Papua - Nova Guiné, Costa Rica e outros países tropicais durante a décima primeira COP em Montreal, 2005 (Silva-Chavez and Petsonk 2006; Schlamadinger *et al.* 2007; Skutsch *et al* 2007; Sedjo and Sohngen 2007). Em dezembro de 2007, a décima terceira reunião da COP ocorrida em Bali, Indonésia, mostrou um caminho para um protocolo do clima pós-Kyoto, a ser iniciado em 2013, o qual enfatiza a necessidade de adotar mecanismos para fornecer incentivos para países em desenvolvimento que reduzam

emissões de carbono do desmatamento. Esta ocasião alavancou um debate internacional sobre como estabelecer métodos de avaliação, regras e meios econômicos para programas REDD.

Uma das principais questões do REDD consiste em medir a contribuição de um projeto (país, estado ou a nível regional) na redução das emissões de carbono por desmatamento e degradação florestal para creditar este esforço. Este tópico é uma das questões mais controversas do REDD porque há muitos meios de medir esse esforço de redução dependendo do histórico de desmatamento da região, estoque de carbono na floresta, e potencial para desmatamento futuro (e.g. Cattaneo, 2008). Por exemplo, o conceito de redução de emissões para um nível abaixo de uma linha de base histórica se aplica bem a países com uma alta taxa histórica de desmatamento e, portanto, com uma grande margem para reduzir suas emissões devido ao desmatamento, tal como o caso do Brasil, com uma linha de base de $19.500 \text{ km}^2 \text{ year}^{-1}$. Em contrapartida, países com grandes extensões de floresta tropical e taxas atualmente baixas de desmatamento (e.g. Peru) não seriam elegíveis à compensação, uma vez que não há esforço adicional para em se reduzir o desmatamento (fig. 19). Entretanto, o desmatamento no Peru provavelmente irá aumentar depois do término da pavimentação da rodovia Interoceânica que liga Brasil ao Oceano Pacífico, assim como outros projetos de infraestrutura que estão em andamento na região (fig. 20). Esses projetos já alavancaram um grande fluxo populacional para o departamento de Madre de Dios no Peru.

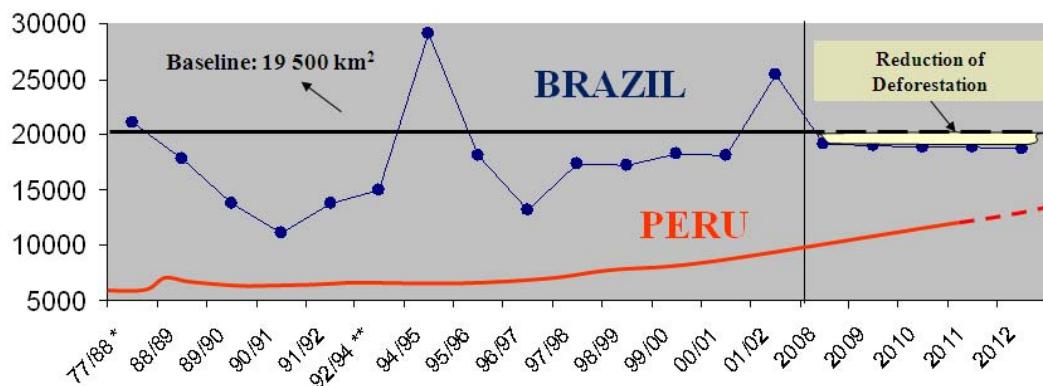


Fig. 19 – Diferentes trajetórias de desmatamento no Brasil e Peru e suas implicações para as propostas de REDD.

Para executar uma avaliação preditiva do potencial para desmatamento futuro sob um cenário Mesmo-de-sempre (Business-as-usual, BAU), projetos pioneiros de REDD, tal como o projeto de conservação PPD do JUMA – Project Design Document (FAS, 2008) adotou o uso de modelos de simulação (Soares-Filho et al., 2006). Neste caso, um conjunto de suposições foi estabelecido para diferenciar o BAU do cenário de governança, tais como pavimentação de rodovias, expansão da agricultura, movimentos populacionais, expansão e consolidação de áreas de proteção e a efetividade de políticas públicas na prevenção do desmatamento. O modelo simula trajetórias de desmatamento sob os cenários modelados e calcula suas respectivas emissões de carbono. Assim, em vez de uma linha de base histórica, o potencial de redução é comparado subtraindo a quantidade acumulada de emissões sob o cenário de governança daquela a partir do cenário BAU até um ano futuro especificado (em geral, adota-se 2050).

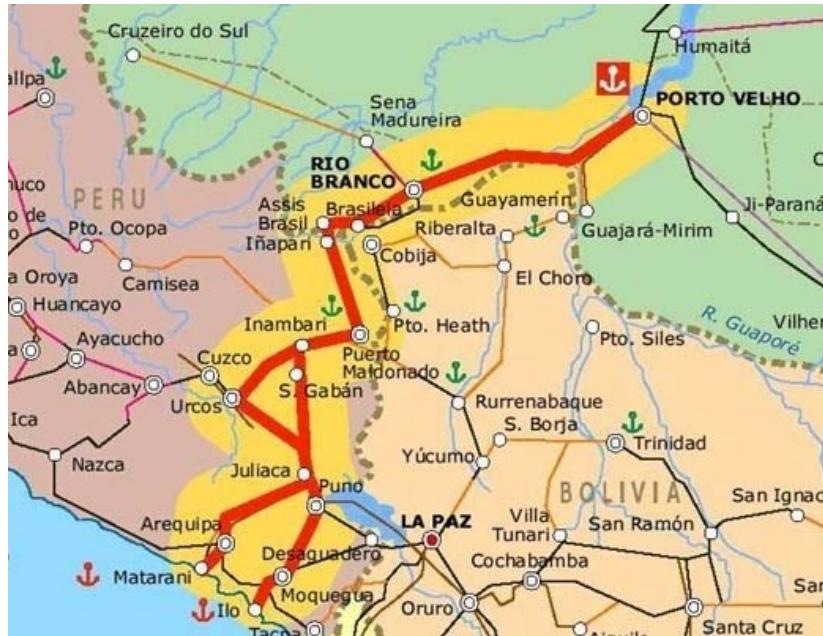


Fig. 20 – Projetos de infraestrutura na região do MAP. MAP é uma sigla para Madre de Dios (Peru), Acre (Brazil) e Pando (Bolivia).

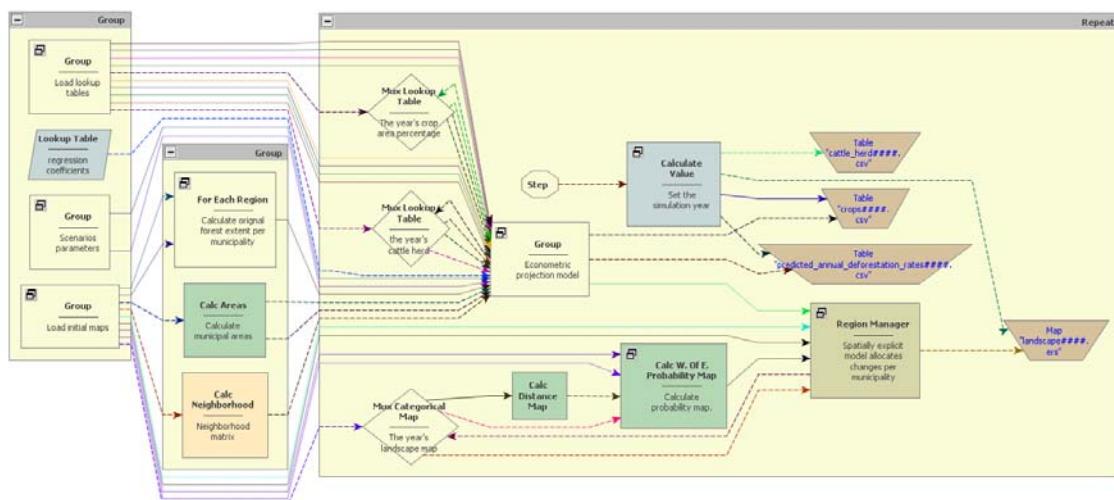
Embora os modelos de simulação desempenhem um importante papel na modelagem dos efeitos de cenários de políticas alternativas de usos da terra na dinâmica da paisagem, esta metodologia deve ser aplicada com cautela aos projetos REDD. A necessidade de se estimar emissões potenciais futuras como um meio de medir a contribuição de uma política pública ou iniciativa de conservação para o REDD, tal como a criação de uma área protegida, está disseminando rapidamente o uso dos modelos de simulação como uma ferramenta para os projetos de REDD. Muitos pacotes comerciais e não-comerciais estão disponíveis para a construção de modelos de simulação espacial. Entretanto, não há solução pronta para um projeto REDD específico (ainda que muitos vendedores digam que sim). Além da simulação dos efeitos de determinantes espaciais na localização do desmatamento (veja lição 7), existe a necessidade de modelar os condicionantes do desmatamento em nível local, regional e mesmo internacional. Isto é bem mais difícil e depende fortemente da disponibilidade de dados socioeconômicos temporais em diferentes escalas, assim como de séries temporais de desmatamento contínuas espacialmente. Tais modelos devem ser construídos a partir da base (i.e. usando abordagens “bottom-up” ao invés de modelos “top-down”). Devem também incorporar nosso conhecimento das causas imediatas e subjacentes do desmatamento, como passar por validação não apenas em termos de sua previsão espacial, mas também considerando o poder de previsão da trajetória do desmatamento recente baseado nas mudanças do contexto socioeconômico e político. Mesmo assim, modelos de simulação não são bolas de cristal, trajetórias futuras modeladas devem ser consideradas apenas como prováveis possibilidades; o que for dito além disso é pura especulação.

Neste contexto, o modelo apresentado aqui tem por objetivo demonstrar as possibilidades do Dinamica EGO na representação do fenômeno de mudanças de uso da terra, não importando sua complexidade. Portanto, no lugar de uma solução pronta para um projeto REDD, o qual provavelmente não irá funcionar mesmo, a plataforma de modelagem do Dinamica EGO fornece meios de materializar, em um domínio computacional, o nosso conhecimento sobre os fenômenos dinâmicos, tais como as maneiras que os contextos políticos e socioeconômicos local, regional e internacional interagem produzindo o desmatamento.

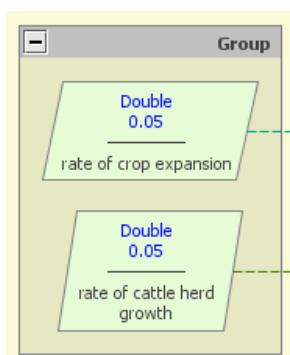
11.1. Desenvolvendo um modelo de projeção econométrica da taxas de desmatamento baseado em mudanças no contexto socioeconômico dos municípios

Neste exemplo, um modelo econométrico é acoplado a um modelo de simulação espacialmente explícito de desmatamento. O modelo de projeção econométrica projeta taxas de desmatamento baseadas em mudanças no contexto socioeconômico dos municípios (Soares-filho *et. al.*, 2008). Uma regressão por defasagem espacial é aplicada para computar a influência de cinco variáveis na trajetória do desmatamento: expansão da área de culturas agrícolas, crescimento dos rebanhos de gado, porcentagem de áreas protegidas, proximidade de estradas pavimentadas e taxas de migração. Uma matriz espacial de vizinhança permite ao modelo incorporar a influência do contexto socioeconômico de municípios vizinhos na previsão de taxas de desmatamento dentro de certo município.

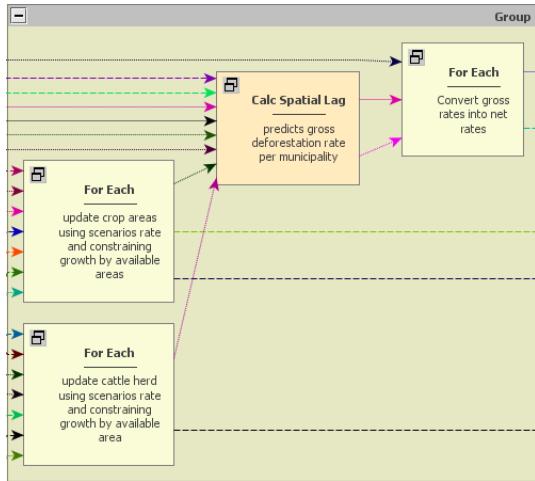
Carregue o modelo “simulate_deforestation_under_socioeconomic_scenarios.xml” from \ Examples\REDD_case_study. Este modelo é composto de três partes principais: os dados de entrada, pré-cálculos, e o modelo de simulação em si.



Nesta versão simplificada do modelo de Soares-Filho *et al.* (2008), o usuário pode modificar o cenário alterando as taxas anuais de expansão da agricultura e aumento de rebanhos de gado, os quais são entradas para o modelo. Outras variáveis podem também ser alteradas editando suas tabelas de entrada.



O *Group* de pré-cálculo calcula a extensão original de floresta por município, a área do município, e a matriz de vizinhança (*Calc neighborhood*) que define quais municípios são vizinhos. Estas serão as entradas para o modelo de projeção. Abra *Group* chamado “Econometric projection model”.

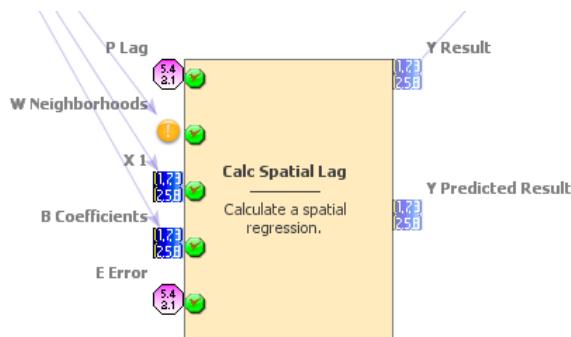


Este *Group* contém três *For Each* e um *Calc Spatial Lag*. Os dois primeiros *For Each* atualizam as tabelas relativas aos rebanhos e áreas de cultivo e calculam suas taxas anuais de mudança, as quais serão entradas para a regressão de defasagem espacial. **DICA:** o *For Each* faz uma varredura pela tabela permitindo a manipulação de seus elementos. Além das tabelas das cinco variáveis independentes, *Calc Spatial Lag* recebe como entrada o coeficiente Lag, a matriz de vizinhança, uma tabela de variável dependente inicial x_1 , e um termo de erro aleatório. Este functor representa uma equação de regressão de defasagem espacial como se segue (Anselin, 2002):

$$y = \rho W y + X\beta + \varepsilon$$

Onde ρ é o coeficiente auto-regressivo, W é uma matriz de vizinhança de primeira ordem, y é a variável dependente, X a matriz de observações para as variáveis independentes, β o vetor de coeficientes de regressão e ε um termo de erro aleatório. Nesta equação, ρW é calculado em um modo iterativo usando médias móveis das respostas de y dos municípios vizinhos. Neste caso, ao invés de um modelo linear clássico, uma regressão de defasagem espacial foi adotada uma vez que o modelo de regressão falhou nos testes de autocorrelação (Anselin, 2002).

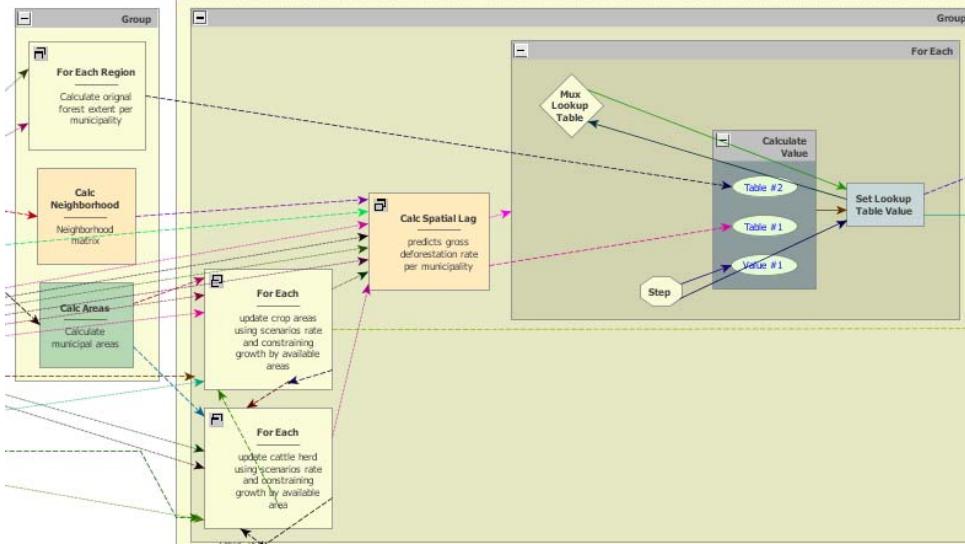
O Dinamica EGO não fornece um método para desenvolver uma regressão de defasagem espacial, mas apenas para um modo de resolver a equação que foi desenvolvida usando *Geoda* (www.geoda.uiuc.edu).



Por fim, o terceiro *For Each* converte as taxas brutas de desmatamento resultantes do *Calc Spatial Lag* em taxas líquidas de desmatamento usando a seguinte fórmula:

$$\text{if } t1[v1] / t2[v1] > 1 \text{ then 1 else if } t1[v1] / t2[v1] < 0 \text{ then 0 else } t1[v1] / t2[v1]$$

onde $t1[v1]$ é a taxa bruta e $t2[v1]$ é a extensão original da floresta de um município.

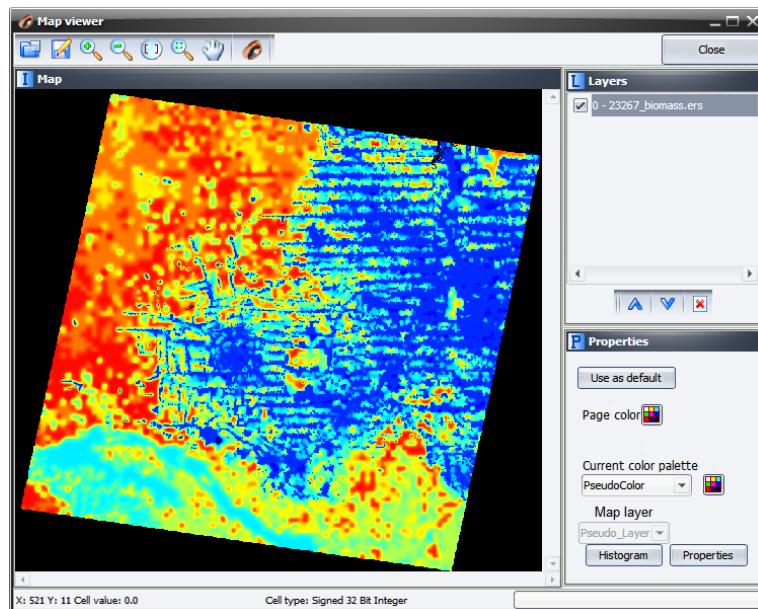


O modelo econométrico passa as taxas de desmatamento previstas para um modelo espacialmente explícito que aloca o desmatamento em cada município, usando a abordagem de sub-regiões da lição 9.2.

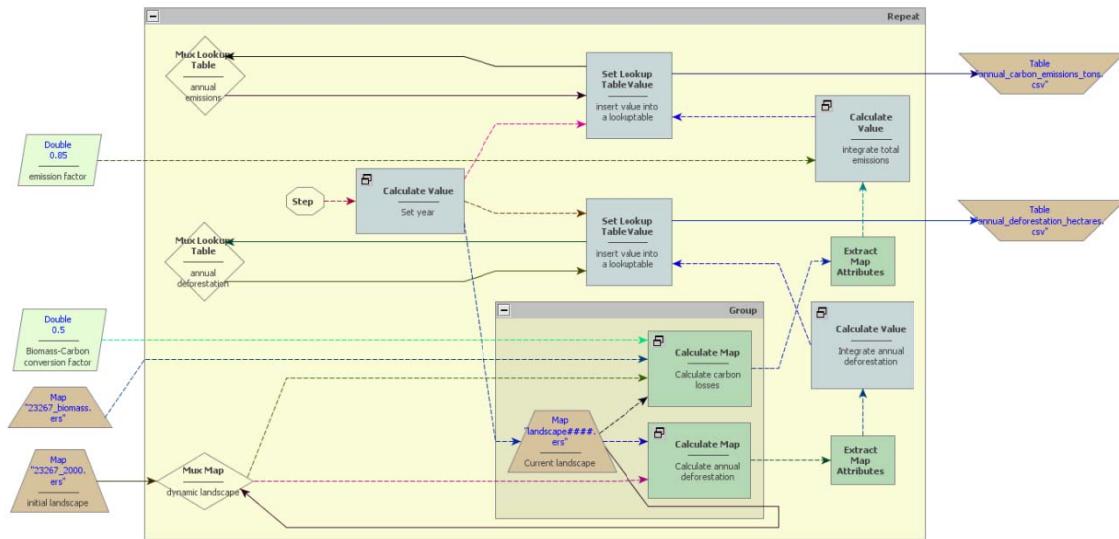
11.2. Desenvolvendo um modelo de contabilidade de carbono

Carregue o modelo “carbon_bookeeping_model.xml” from \ Examples\REDD_case_study

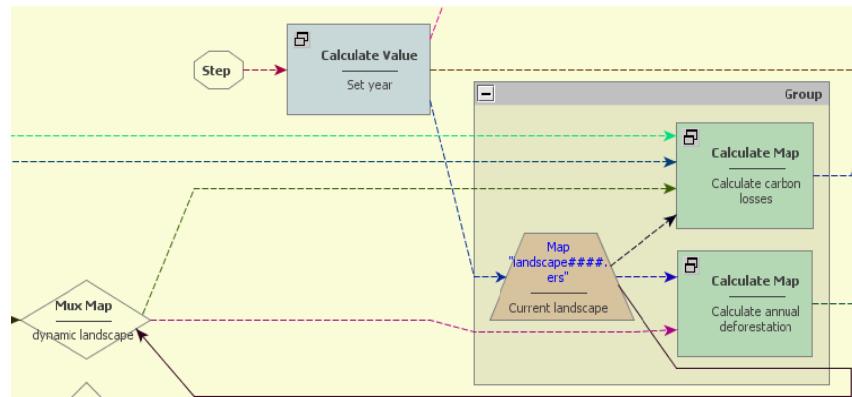
Este modelo calcula as emissões anuais de carbono identificando o desmatamento anual e, em seguida, sobrepondo essas áreas em um mapa de carbono da biomassa florestal – figura abaixo (Saatchi *et al.*, 2007), e assumindo que o conteúdo de carbono representa 50% da biomassa florestal (Houghton *et al.*, 2001) e que 85% do carbono contido nas árvores é liberado na atmosfera pelo desmatamento (Houghton *et al.*, 2000).



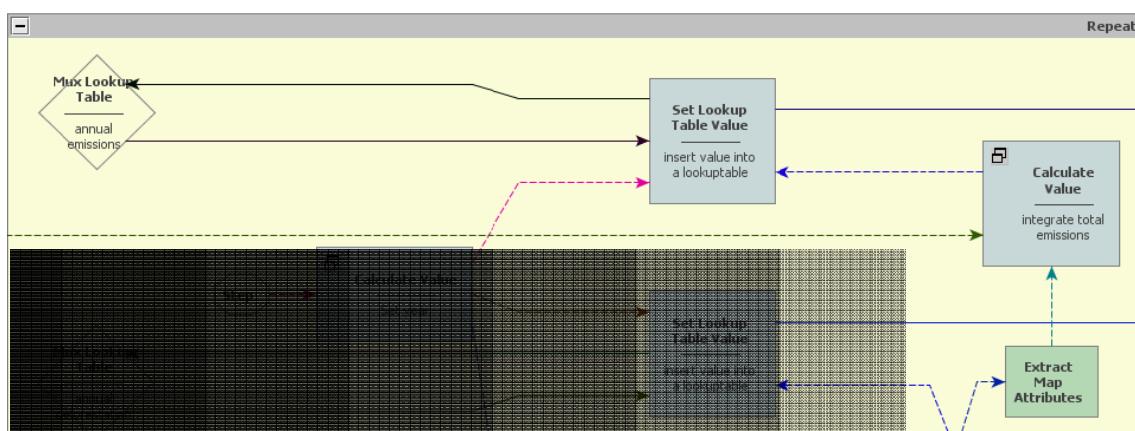
Para calcular o desmatamento anual, o modelo compara, em cada passo de tempo, o mapa atual de uso da terra com o anterior. Dinamica EGO permite o carregamento de múltiplos mapas através do uso do *Load Map* dentro do *Repeat*, usando *Step* para encontrar o arquivo cujo nome tem o número do passo do modelo como sufixo. Observe que, neste caso, o sufixo tem 6 dígitos para poder comportar o ano de simulação (2002-2020).



Um *Load Map* é colocado dentro de um *Group* para assegurar uma ordem apropriada de execução. O mapa anterior de uso da terra é mantido em um *Mux Map*, de forma que ambos os functores *Calculate Map* neste container recebem o mapa anterior de uso da terra como **i1** e o atual como **i2**.



Depois que as células do desmatamento anual são identificadas, o modelo obtém os estoques de biomassa correspondentes no mapa de biomassa e os converte em carbono e, em seguida, em emissões. O *Extract Map Attribute* é aplicado para calcular a quantidade total de células e o *Calculate Value* integra aqueles valores em uma base anual. Seu resultado é passado para o *Set Lookup Table* que atualiza uma tabela com as emissões anuais de carbono (Fig. 21).



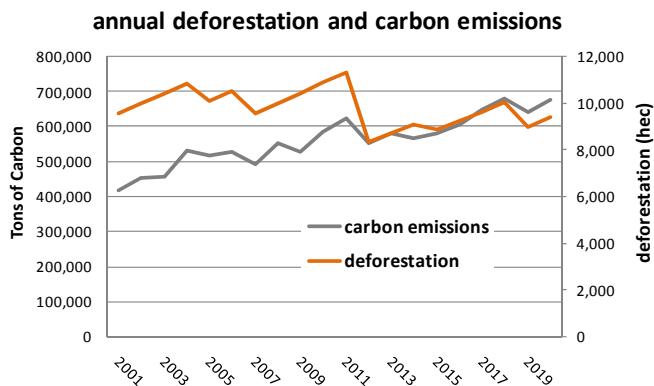


Fig. 21 – Desmatamento anual e emissões calculadas para a região do modelo.

Observações Finais

Parabéns! Você completou este guia prático. Que canseira! Outros exemplos e textos sobre Dinamica EGO e suas aplicações estão disponíveis na pasta Examples. Nós esperamos que você tenha compreendido o raciocínio básico por trás da construção de modelos com Dinamica EGO e possa vislumbrar, a partir de agora, suas enormes possibilidades para a modelagem ambiental. Esta é apenas a ponta do iceberg, agora cabe a você explorar esse vasto ambiente destinado ao criativo e engenhoso desenvolvimento de modelos ambientais.

Dinamica EGO é um software gratuito, sem fins comerciais (veja o copyright abaixo). Vários programas de pesquisa conduzidos pelo CSR/UFMG têm financiado seu desenvolvimento. Nós somos gratos a todos os patrocinadores que, direta ou indiretamente, contribuíram para seu atual estado-da-arte. Veja a lista de patrocinadores no nosso sítio de internet [www.csr.ufmg.br/dinamica](http://www csr ufmg br/dinamica).

Nós estamos comprometidos com o apoio a estudantes e pesquisadores interessados em seu uso. Para obter informações adicionais, sinta-se livre para entrar em contato - dinamica@csr.ufmg.br. O laboratório CSR, na UFMG, possui um programa para visita de estudantes

Dinamica EGO Copyright

Copyright(c) 1998-2009 **Centro de Sensoriamento Remoto / Universidade Federal de Minas Gerais** - Brazil. Todos os direitos reservados.

O Produto é licenciado a você em uma base “Como está”, sujeito aos termos e condições deste acordo, apenas para seu uso pessoal e não comercial. Portanto, ele não deve ser transferido no todo ou em parte, a terceiros que sejam instituições comerciais, ou que façam uso comercial dele. Instituição comercial é definida, aqui, como qualquer companhia ou organização com interesses de lucro, uso comercial é definido como qualquer uso dentro de uma instituição comercial, qualquer uso com interesse de lucro ou qualquer uso em cooperação de duas ou mais partes, tendo, no mínimo, uma delas interesse de lucro ou sendo uma instituição comercial. Outros direitos além daqueles claramente garantidos a você relacionados abaixo, nenhum outro direito é garantido a você. Sem limitar o que foi dito anteriormente, você não poderá: (a) modificar ou criar qualquer trabalho derivado deste Produto ou documentação; (b) decompilar, desmontar, engenharia reversa, ou qualquer outra tentativa de obter o código fonte do Produto; ou (c) redistribuir, atrapalhar vender, alugar, sublocar, sublicenciar, ou qualquer outra transferência de direitos do Produto sem a permissão prévia por escrito do **CSR/UFMG**.

12. Referências

- Agterberg, F.P. and Bonham-Carter, G.F. Deriving weights of evidence from geoscience contour maps for the prediction of discrete events. *XXII Int. Symposium AP-COM*, 381-395. (1990).
- Anselin, L. *Spatial Externalities, Spatial Multipliers and Spatial Econometrics*. (University of Illinois, Urbana-Champaign, 2002).
- Bonham-Carter, G. *Geographic information systems for geoscientists: modeling with GIS*. (Pergamon, New York, 1994) 398 pp.
- Burrough, P. A. *Principles of Geographical Information Systems for Land Resources Assessment*. (Clarendon Press, Oxford, 1986).
- Cattaneo, A. *How to Distribute REDD Funds Across Countries? A Stock-Flow Mechanism*. Submission to the United Nations Framework Convention on Climate Change regarding AWG-LCA (FCCC/AWGLCA/2008/L.7) (2008).
- Costa, M. H., Botta, A. and Cardille, J. A. Effects of large-scale changes in land cover on the discharge of the Tocantins river, Southeastern Amazonia. *Journal of Hydrology* **283**, 206-217 (2003).
- Costanza, R. Model goodness of fit: a multiple resolution procedure. *Ecological Modelling*, **47**, 199-215 (1989).
- FAS (Fundação Amazônia Sustentável). *The Juma Sustainable Development Reserve Project: Reducing Greenhouse Gas Emissions from Deforestation in the State of Amazonas, Brazil. For validation at Climate Community and Biodiversity alliance (CCBA)*. [online] <http://www.climate-standards.org/pdf/release_juma_english_v_1_0_3.pdf> (2008).
- Fearnside, P. M. Environmental Services as a Strategy for Sustainable Development in Rural Amazonia. *Ecological Economics* **20** 53-70 (1997).
- Geist, H. J. and Lambin, E. F. *What Drives Tropical Deforestation? A Meta-Analysis of Proximate and Underlying Causes of Deforestation Based on Subnational Case Study Evidence*. Belgium, LUCC International Project Office, LUCC Repo6rt Series, 4. 136pp. [online] <<http://www.geo.ucl.ac.be/LUCC/lucc.htm>> (2001).
- Goodacre C. M., Bonham-Carter G. F., Agterberg, F. P. and Wright D. F. A statistical analysis of spatial association of seismicity with drainage patterns and magnetic anomalies in western Quebec. *Tectonophysics* **217**, 205-305 (1993).
- Hagen, A. Fuzzy Set Approach to Assessing Similarity of Categorical Maps. *International Journal of Geographical Information Science*, **17**, 235-249 (2003).
- Houghton, R.A. et al. Annual fluxes of carbon from deforestation and regrowth in the Brazilian Amazon. *Nature* **403**, 301-304 (2000).
- Houghton, R. A., Lawrence, K. T., Hackler, J. and Brown, L. S. The spatial distribution of forest biomass in the Brazilian Amazon: a comparison of estimates. *Global Change Biology* **7**, 731-746 (2001).
- Houghton, R.A. in *Tropical Deforestation and Climate Change*. (eds Moutinho and S. Schwartzman) (IPAM and ED, Belém, 2005).
- Hirsch, A. I., Little, W. S., Houghton, R. A., Scott, N. A. and White, J. D. The net carbon flux due to deforestation and forest re-growth in the Brazilian Amazon: analysis using a process-based model. *Global Change Biology* **10**, 908-924 (2004).
- INPE (Instituto Nacional de Pesquisas Espaciais). *Monitoramento da Floresta Amazônica Brasileira por Satélite - Projeto PRODES*. [online] <<http://www.obt.inpe.br/prodes>> (2007)
- McGarigal, K. and Marks, B.J. FRAGSTATS: Spatial pattern analysis program for quantifying landscape structure. PNW-GTR-351. U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station, Portland. (1995).
- Metzger, J.P. How to deal with non-obvious rules for biodiversity conservation in fragmented landscapes? *The Brazilian Journal of Nature Conservation* **4**, 125-139 (2006).
- Moutinho, P. and S. Schwartzman. *Tropical Deforestation and Climate Change*. (IPAM and ED, Belém, 2005).

- Merry, F., Soares-Filho, B.S., Nepstad, D., Amacher, G. and Rodrigues, H. Balancing Conservation and Economic Sustainability: The future of the Amazon timber industry. *Environmental Management* (2009). doi: 10.1007/s00267-009-9337-1
- Pontius, R.G. Jr., Statistical Methods to Partition Effects of Quantity and Location During Comparison of Categorical Maps at Multiple Resolutions. *Photogrammetric Engineering and Remote Sensing* **68**, 1041-1049 (2002).
- Power, C., Simms, A. and White, R. Hierarchical fuzzy pattern matching for the regional comparison of Land Use Maps. *International Journal of Geographical Information Science* **15**, 77-100 (2001).
- Saatchi, S. S., Houghton, R. A., Dos Santos Alvala, R. C., Soares, Z. J. V. and Yu, Y. Distribution of aboveground live biomass in the Amazon basin. *Global Change Biology* **13**, 816–837 (2007).
- Schlundadinger, B., Johns, T., Ciccarese, L., Braun, M., Sato, A., Senyaz, A., Stephens, P., Takahashi, M. and Zhan, X. Options for including land use in a climate agreement post-2012: improving the Kyoto Protocol approach. *Environmental Science and Policy* **10**, 295-305 (2007).
- Sampaio, G., Nobre, C., Costa, M. H., Satyamurty, P., Soares-Filho, B. S., Cardoso, M. Regional climate change over eastern Amazonia caused by pasture and soybean cropland expansion. *Geophysical Research Letters* **34**, 1-7 (2007). doi: 10.1029/2006GL027002
- Santilli, M., Moutinho, P., Schwartzman, S., Nepstad, D. C., Curran, L., and Nobre, C. Tropical deforestation and the Kyoto Protocol: an editorial essay. *Climatic Change* **71**, 267-276 (2005).
- Schneider, E. K., Fan, M; Kirtman, B. P. and Dirmeyer, P. Potential effects of Amazon deforestation on tropical climate. *Cola Technical Report* **226**, 1-41 (2006).
- Sedjo, R.A., B. Sohngen. Carbon credits for avoided deforestation. Washington, D.C, *Resources for the Future* (2007)
- Silva-Chavez, G. and Petsonk, A. Rainforest credits. *Carbon Finance* **6**, 18 (2006).
- Skutsch, M., Bird, N., Trines, E., Dutschke, M., Frumhoff, P., de Jong, B. H. J., van Laake, P., Masera, O., and Murdiyarso, D. Clearing the way for reducing emissions from tropical deforestation. *Environmental Science and Policy* **10**, 322-334 (2007).
- Soares-Filho, B. S., Pennachin, C. L., Cerqueira, G. DINAMICA – a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. *Ecological Modelling* **154**, 217-235 (2002).
- Soares-Filho, B. S., Corradi, L., Cerqueira, Araújo, W. Simulating the spatial patterns of change through the use of the dinamica model. In *Simpósio Brasileiro de Sensoriamento Remoto*, **11**, 2003, BH, INPE, 721-728 (2003).
- Soares-Filho, B. S., Alencar, A., Nepstad, D., Cerqueira, G., Vera-Diaz, M., Rivero, S., Solórzano, L. and Voll, E. Simulating the response of land-cover changes to road paving and governance along a major Amazon highway: the Santarém-Cuiabá corridor. *Global Change Biology* **10**, 745-764 (2004).
- Soares-Filho, B. S., Nepstad, D., Curran, L., Voll, E., Cerqueira, G., Garcia, R. A., Ramos, C. A., McDonald, A., Lefebvre, P. and Schlesinger, P. Modeling Conservation in the Amazon Basin. *Nature* **440**, 520-523 (2006).
- Soares-Filho, B.S., Garcia, R. A., Rodrigues, H., Moro, S. and Nepstad, D. Nexos entre as dimensões socioeconômicas e o desmatamento: A caminho de um modelo integrado. In *Amazônia. Natureza e Sociedade em Transformação*. (eds Batistella, M., Alves, D. and Moran, E.) (Edusp, São Paulo, 2008).
- Teixeira, A. M., Soares-Filho, B.S., Freitas, S. and Metzger, J.P.W. Modeling Landscape dynamics in the Atlantic Rainforest domain: Implications for conservation. *Forest Ecology and Management* **257**, 1219–1230 (2009).

13. Lista de functores

	pg.
<i>Add Change Matrix</i>	84, 85, 93
<i>Calc Areas</i>	51, 52, 56
<i>Calc Change Matrix</i>	57, 71, 73, 93
<i>Calc Cost Map</i>	33, 36, 37, 38, 39, 46, 47
<i>Calc Distance Map</i>	65, 71, 73, 88, 96
<i>Calc Distance To Feature Map</i>	42, 43, 44, 45
<i>Calc Neighborhood</i>	104, 107
<i>Calc Patch Label Map</i>	51, 52, 56
<i>Calc Pathway Map</i>	33, 36, 37, 38, 39
<i>Calc Reciprocal Similarity Map</i>	75, 77, 79
<i>Calc Spatial Lag</i>	104, 108
<i>Calculate Categorical Map</i>	42, 43, 45, 53, 55, 75, 100
<i>Calculate Map</i>	6, 14, 15, 16, 23, 24, 25, 26, 32, 33, 35, 36, 37, 39, 41, 43, 44, 45, 46, 48, 49, 50, 52, 53, 56, 77, 79, 99, 102, 110
<i>Calculate Value</i>	23, 26, 28, 54, 77, 80, 86, 88, 110
<i>Calc WOFE Probability Map</i>	57, 65
<i>Determine Transition Matrix</i>	60
<i>Determine Weights Of Evidence Coefficients</i>	57, 65, 67
<i>Determine Weights Of Evidence Correlation</i>	70
<i>Determine Weights Of Evidence Ranges</i>	57, 65
<i>Expander</i>	57, 60, 73, 83, 84, 85, 88, 90, 99, 100
<i>Extract Map Attributes</i>	24, 26, 54
<i>For</i>	79, 80
<i>For Each</i>	56, 108
<i>For Each Category</i>	30, 31, 32, 53, 54, 97, 98
<i>Group</i>	40, 50, 51, 56, 77, 80, 86, 98, 103, 104, 108, 110
<i>If Not Then</i>	77
<i>If Then</i>	77
<i>Int</i>	98
<i>Join Map</i>	77
<i>Load Categorical Map</i>	24, 25, 36, 41, 44, 71, 95
<i>Load Lookup Table</i>	71

<i>Load Map</i>	14, 15, 16, 21, 24, 25, 31, 34, 36, 53, 66, 71, 109, 110
<i>Load Weights</i>	14, 67, 70, 71, 94, 97
<i>Log Policy</i>	6, 103
<i>Lookup Table</i>	34, 35
<i>Modulate Change Matrix</i>	84, 85, 93
<i>Mux Categorical Map</i>	71, 72, 88, 98
<i>Mux Lookup Table</i>	24, 27, 28, 54, 55, 80
<i>Mux Map</i>	110
<i>Mux Value</i>	56
<i>Name</i>	71
<i>Name Map</i>	65, 66, 68, 71
<i>Number Map</i>	14, 15, 16, 24, 25, 32, 35, 41, 42, 43, 47, 49, 52, 53, 55, 65
<i>Number Table</i>	26, 35, 51, 54, 54
<i>Number Transition Matrix</i>	93, 94
<i>Number Value</i>	25, 86
<i>Number Weights</i>	94
<i>Patcher</i>	57, 60, 72, 73, 82, 84, 85, 88, 99, 100
<i>Regional Categorical Map</i>	97, 98
<i>Regionalize Categorical Map</i>	97
<i>Regionalize Map</i>	32, 33, 53, 97
<i>Region Manager</i>	30, 31, 32, 53, 55, 56, 96, 97
<i>Repeat</i>	24, 25, 27, 29, 30, 31, 71, 73, 74, 80, 86, 94, 96, 109
<i>Save Lookup Table</i>	29
<i>Save Map</i>	14, 15, 16, 22, 29, 31, 35, 36, 37, 38, 41, 42, 43, 44, 47, 48, 49, 52, 53, 66, 71, 72, 74, 86
<i>Save Weights</i>	69
<i>Select Transition Matrix</i>	93, 94
<i>Select Weights</i>	93, 94
<i>Set Lookup Table Value</i>	24, 27, 28, 29, 30, 81
<i>Step</i>	24, 25, 26, 28, 32, 53, 54, 79, 80, 86, 97, 109

Dinamica EGO tem sido aplicado a numerosos estudos ambientais, incluindo a modelagem de desmatamento na Amazônia, desde escala local a continental, mudanças na cobertura e uso do solo na Floresta Atlântica e na floresta seca do México, dinâmica urbana, exploração madeireira na Amazônia, risco e espalhamento de fogo, análise de custos de oportunidade da substituição de atividades agrícolas e florestais por conservação de floresta, o papel das áreas protegidas na redução das emissões de carbono na Amazônia Brasileira, co-benefícios de REDD nas cabeceiras do Xingu e uma proposta para o fim do desmatamento na Amazônia Brasileira. O objetivo deste guia prático consiste em introduzir o usuário às vastas possibilidades do Dinamica EGO para o desenho criativo de modelos que podem verdadeiramente representar a complexidade de fenômenos geográficos.